Rich Communication Suite 5.2 Advanced Communications Services and Client Specification

Version 5.0

07 May 2014

This is a Non-binding Permanent Reference Document of the GSMA

Security Classification: Non-confidential

Copyright Notice

Disclaimer

Antitrust Notice

Table of Contents

1 Introduction

1.1 RCS 5 Principles and Vision

RCS (Rich Communication Suite) 5.2 provides a framework for discoverable and interoperable advanced communication services and detailed specifications for a basic set of advanced communication services. RCS 5.2 builds on the fundamentals from previous RCS versions (such as RCS 5.1, see [RCS51]) that are succeeded by this specification.

As indicated in Figure 1, the set of services specified in RCS 5.2 includes all services from RCS 5.1. RCS 5.2 extends this basis with new services (indicated in **bold** in Figure 1) and provides enhancements for existing services (indicated in *italics* in Figure 1). All these services can be deployed using a variety of clients on access networks that can be Service Provider controlled or not.

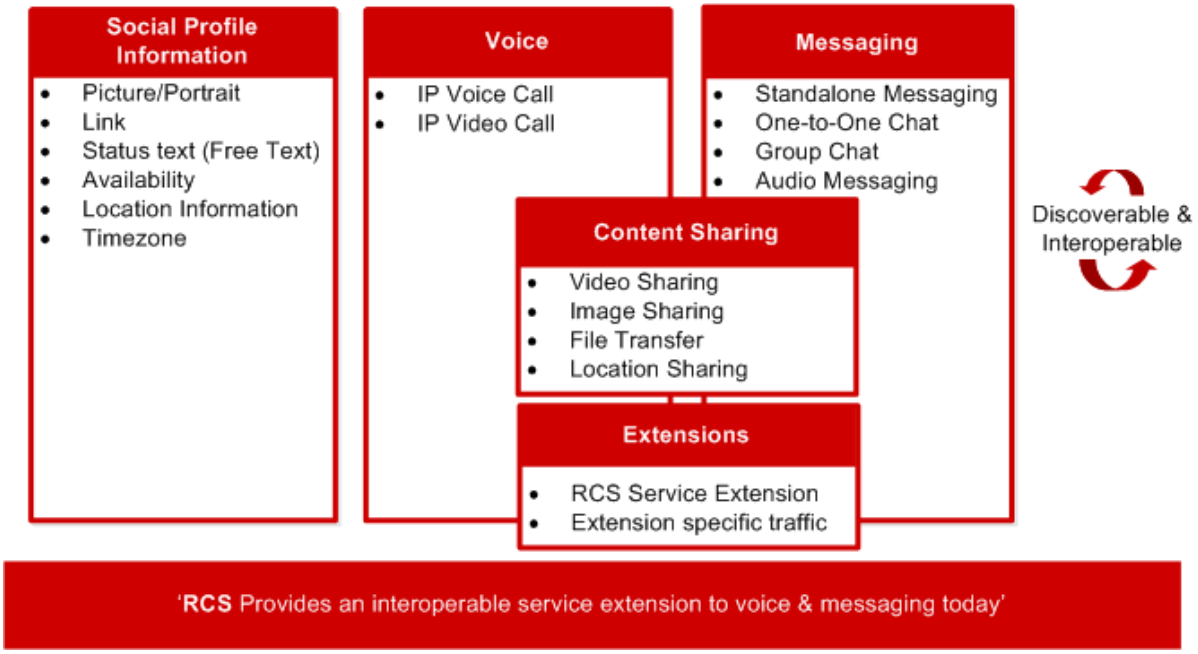


Figure 1: RCS Positioning

As a headline, RCS provides an "interoperable extension to voice and messaging today". The services are designed to run over data networks and can stand alone (e.g. share a picture from the media gallery) or be used in combination with a voice call (e.g. see-what-I-see video).

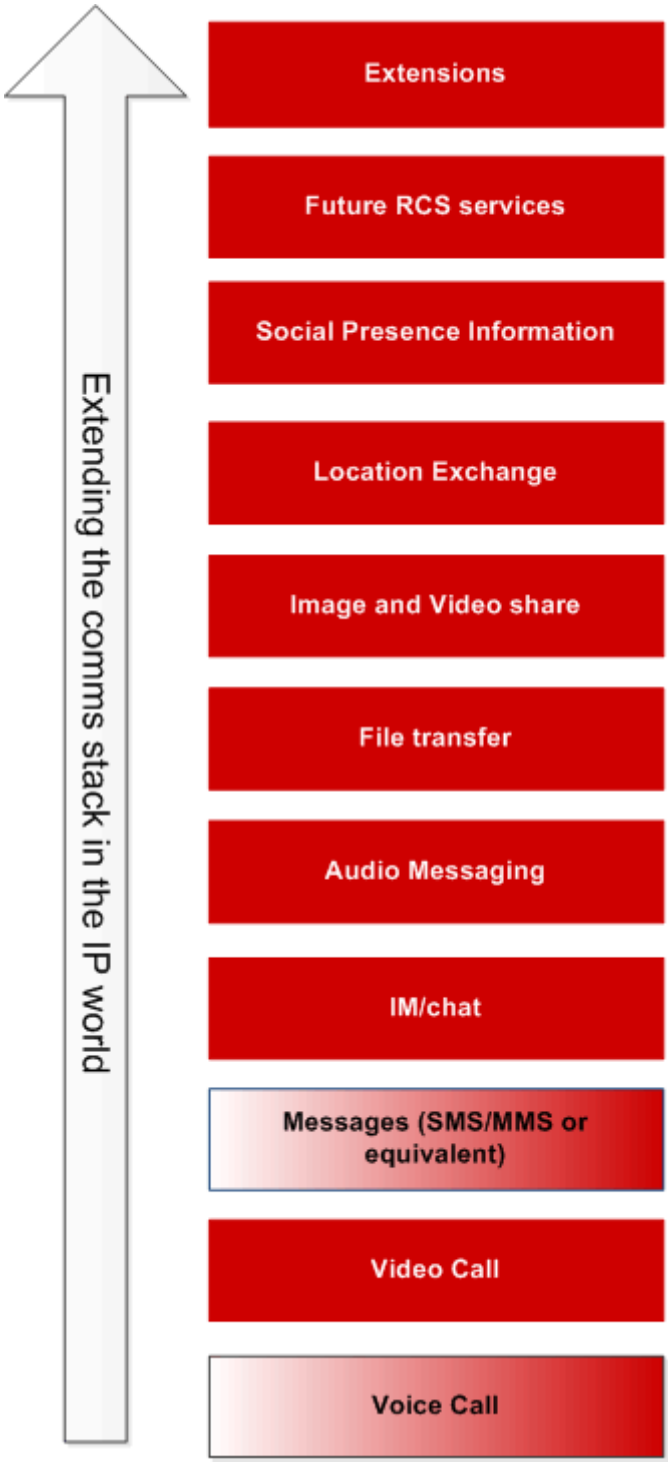


Figure 2: RCS Industry Proposition

As indicated in section 1.2.2, a Service Provider may choose not to deploy all services defined as part of RCS 5.2; however when deploying an individual RCS 5.2 service it will be interoperable with other Service Providers deploying the same service. This also means that even if this specification offers different deployment options to accommodate for different market realities, full interoperability between those deployments is provided for each corresponding service and for the RCS discovery framework. As a consequence of the choices in this specification, RCS 5.1 is itself a subset of the RCS 5.2 functionality.

The cornerstone mechanism that enables RCS is a service or capability discovery framework. For example, when a user scrolls through their address book, they will see their contacts with the RCS services that are available to communicate.

This mechanism is implemented either using the Session Initiation Protocol's (SIP) OPTIONS request or using a Presence-based solution defined in RCS Release 1-4. Both will result in one of three types of response:

1. The contact is registered for service resulting in the contact's current service capabilities being received and logged, or,
2. The contact is not registered (they are provisioned but not registered)
3. The contact is not found (they are not provisioned for service).

This discovery mechanism is important since it ensures User A can determine what services are available before communicating and allows Service Providers to roll-out new agreed services based on their own deployment schedule or market requirements. These same mechanisms can be used to initially discover (and/or periodically check) the service capabilities of all the contacts within an address book when the user first registers for the service.

1.2 Scope

This document extends the core principles and services framework from the initial set of functionalities defined in RCS 5.0 and RCS 5.1. The framework is designed to be extensible and to support new services going forward.

This document focuses mainly on the User Network Interface (UNI) which to a large extent also determines the Network-Network Interface (NNI). This document also specifies how networks who may choose a different set of deployment options (from the ones described) can work with each other. The interconnect-specific aspects of the NNI are described in a separate document (see [PRD-IR.90]).

It should be noted that the aim of this document is to only specify functionality that can be validated in standard compliant Internet Protocol (IP) Multimedia Subsystem (IMS) pre-production and production environments without major customisation or changes. Service Providers can still introduce customizations and changes to optimise or differentiate their networks however.

It should be noted that all text describing the User Experience (UX), pictures and flow diagrams are for informative purposes only.

1.2.1 Original Equipment Manufacturer (OEM) Integration

This specification is independent from any specific device operating system and is not intended to prescribe the supplier user experience. However, where appropriate key service logic is illustrated through wireframes to aid the reader. It is expected that each device or client supplier will map the basic service principles defined in this document within their own products and drive innovative and differentiated experiences.

1.2.2 Conformance

For terminals, the minimum conformance to the RCS 5.2 specification can be achieved by a terminal providing the necessary functionality to support the RCS framework, including the capability and new user discovery mechanism (covered in detail in section 2) and one or more of the services specified in detail in section 3. Support for multiple services is optional, however is highly recommended. These conformance criteria ensure that RCS can target low-end devices and therefore boost the market penetration curve.

For networks, the conformance criteria are similar. The framework should be supported including the measures to provide compatibility with all other deployed networks and at a

minimum one of the services should be supported. Also, the network should prevent non-compliant clients from connecting to the network or affecting the UNI to a compliant client or the NNI to a compliant network.

1.3 Definition of Terms

Term	Description
2G	2nd Generation of Global System for Mobile Communications (GSM)
ACK	Acknowledgement
ACL	Access Control List
ADSL	Asymmetric Digital Subscriber Line
ALG	Application Layer Gateway
AMR	Adaptive Multi-Rate
APN	Access Point Name
AP	Authentication Proxy
API	Application Programming Interface
AS	Application Server
ASAP	As Soon As Possible
ASO	Arbitrary Slice Ordering
AVC	Advanced Video Codec
BA	Broadband Access
B2BUA	Back-to-Back User Agent
bool	Boolean
BP	Baseline Profile
BPEF	Blacklist Policy Enforcement Function
bps	Bits per second (used with Mbps: Mega-, kbps: kilo-)
CA	Certification Authority
CAB	Converged Address Book
CBP	Constrained Baseline Profile
CCW	Counter-Clockwise
CPIM	Common Profile for Instant Messaging
CPM	Converged IP Messaging
CRLF	Carriage Return Line Feed
CS	Circuit Switched
CSFB	Circuit Switched FallBack
CVO	Coordination of Video Orientation
CW	Clockwise
DNS	Domain Name System
DNS SRV	Domain Name System Service record
DRX	Discontinuous Reception
DTM	Dual Transfer Mode

DTX	Discontinuous Transmission
e2ae	end-to-access edge
e2e	end-to-end
EAB	Enhanced Address Book
eIMS-AGW	Enhanced IP Multimedia Subsystem-Access GateWay
EOF	End Of File
eP-CSCF	Enhanced Proxy-Call Session Control Function
EPSG	European Petroleum Survey Group
EUCR	End User Confirmation Request
Extension	Piece of software (e.g. add-on, app, etc.) installed on top of an RCS Client that makes use of the RCS infrastructure to change or enhance the user experience or bring extra functionality to the service via the existing or via a new, separate user interface.
FIFO	First IN First Out
FIR	Full Intra Request
FMO	Flexible Macroblock Ordering
FQDN	Fully Qualified Domain Name
FTF	File Transfer Function
FTTH	Fibre To The Home
GAA	Generic Authentication Architecture
GBA	Generic Bootstrapping Architecture
GBR	Guaranteed Bitrate
GGSN	Gateway General Packet Radio Service Support Node
GIBA	General Packet Radio Service -IMS-Bundled Authentication
GIF	Graphics Interchange Format
GML	Geography Markup Language
GMLC	Gateway Mobile Location Centre
GPRS	General Packet Radio Service
GPS	Global Positioning System
GRUU	Globally Routable User agent URI
GSM	Global System for Mobile Communications
GSMA	GSM Association
GSO	Group State Object
HPLMN	Home Public Land Mobile Network
H-SLP	Home SUPL Location Platform
HSPA	High Speed Packet Access
HSS	Home Subscriber Server
HTTP	Hyper-Text Transfer Protocol
HTTPS	Hyper-Text Transfer Protocol Secure
HW	HardWare

IARI	IMS Application Reference Identifier
I-CSCF	Interworking Call Session Control Function
ICSI	IMS Communication Service Identifier
ID	IDentifier
IETF	Internet Engineering Task Force
IM	Instant Messaging. The term chat is also applied in this document to the same concept.
IMAP	Internet Message Access Protocol
IM-AS	Instant Messaging Application Server NOTE: This equivalent terminology for Messaging Server is used in some of the figures
IMDN	Instant Message Disposition Notification
IMEI	International Mobile Station Equipment Identity
IMPI	Internet Protocol Multimedia Subsystem Private Identity
IMPU	Internet Protocol Multimedia Subsystem PUblic identity
IMS	Internet Protocol Multimedia Subsystem
IMSI	International Mobile Subscriber Identity
IMS AKA	IMS Authentication and Key Agreement
Int	Integer
IP	Internet Protocol
IPsec	Internet Protocol Security
IP-SM-GW	Internet Protocol Short Message Gateway
IPX	Internet Protocol Packet eXchange
ISIM	Internet Protocol Multimedia Services SIM
ISF	Interworking Selection Function
IWF	InterWorking Function
JPEG	Joint Photographic Experts Group
KB	KiloByte (i.e. 1024 bytes)
kB	Kilobyte 1 kilobyte = 10^3bytes = 1000bytes.
LBS	Location Based Services
LSB	Least Significant Bit
LTE	Long Term Evolution
MAP	Mobile Application Part
Messaging Server	A server providing support for the standalone messaging service (see section 3.2) according to [RCS5-CPM-CONVFUNC-ENDORS] and/or Chat (see sections 3.3 and 3.4) according to [RCS5-SIMPLEIM-ENDORS] or [RCS5-CPM-CONVFUNC-ENDORS]
MIME	Multipurpose Internet Mail Extensions
MLP	Mobile Location Protocol
MMS	Multimedia Message Service

MMS-C	Multimedia Messaging Service Centre
MMTEL	MultiMedia TELephony
MNO	Mobile Network Operator
MO	Management Object
MO-SMS	Mobile Originated Short Message Service
MPEG	Moving Pictures Experts Group
MSB	Most Significant Bit
MSISDN	Mobile Subscriber Integrated Services Digital Network Number
MSRP	Message Session Relay Protocol
MSRPoTLS	Message Session Relay Protocol over Transport Layer Security
MTU	Maximum Transmission Unit
NAL	Network Abstraction Layer
NAT	Network Address Translation
NGBR	Non-Guaranteed Bitrate
NNI	Network Network Interface
NW	NetWork
OEM	Original Equipment Manufacturer
OMA	Open Mobile Alliance
OMA-CP	Open Mobile Alliance Client Provisioning
OMA-DM	Open Mobile Alliance Device Management
OS	Operating System
OTP	One Time Password
PCO	Protocol Configuration Options
P-CSCF	Proxy-Call Session Control Function
PC	Personal Computer
PCC	Personal Contact Card
PDP	Packet Data Protocol
PDF	Portable Document Format
PIDF	Presence Information Data Format
PKI	Public Key Infrastructure
PNB	Personal Network Blacklist
PNG	Portable Network Graphics
PPS	Picture Parameter Set
PRD	Permanent Reference Document
PS	Packet Switched
PSTN	Public Switched Telephone Network
QCI	Quality of Service Class Identifier
QoS	Quality of Service

RADIUS	Remote Authentication Dial In User Service
RAN	Radio Access Network
RCS	Rich Communication Suite
RCS Group Chat ID	A globally unique identifier that uniquely identifies a Group Chat and that is created when the group chat is first started and preserved over Group Chat restarts. The Group Chat ID is transported as the Contribution-ID header field in a SIMPLE-IM based communication and as the Conversation-ID header field when the communication is CPM based.
RCS User	An end user that has device or client (and the corresponding Service Provider subscription) supporting the RCS capability exchange framework and at least one of the services defined in the current specification.
RFC	Request For Comments
RLC	Radio Link Control
RLS	Resource List Server
RR	Receiver Report
RRAM	RCS Recorded Audio Message
RS	Redundant Slices
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
SASL	Simple Authentication and Security Layer
SBC	Session Border Controller
S-CSCF	Serving Call Session Control Function
SDP	Session Description Protocol
SDES	Session Description Protocol Security Descriptions for Media Streams
SET	Secure User Plane Location Enabled Terminal
SGs interface	3GPP defined reference point between the Mobility Management Entity and the Mobile Switching Centre
SIM	Subscriber Identity Module
SIMPLE	Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions
SIO	Session Info Object
SIP	Session Initiation Protocol
SIPoTLS	Session Initiation Protocol over Transport Layer Security
SLA	Service Level Agreement
SMPP	Short Message Peer-to-Peer
SMS	Short Message Service
SMS-C	Short Message Service Centre
SMSoIP	Short Message Service over Internet Protocol
SP	Service Provider
SPI	Social Presence Information
SPS	Sequence Parameter Set

SR	Sender Report
SRTP	Secure Real-time Transport Protocol
SR-VCC	Single Radio Voice Call Continuity
SSO	Single Sign On
STAP-A	Single-Time Aggregation Packet type A
STUN	Simple Traversal of User Datagram Protocol through Network Address Translations
SUPL	Secure User Plane Location
SW	SoftWare
TCP	Transmission Control Protocol
tel URI	telephone Uniform Resource Identifier
TID	Transaction IDentifier
TLS	Transport Layer Security
TON	Type Of Number
TPDU	Transfer Protocol Data Unit
UA	User Agent
UC	Use Case
UCS2	2-byte Universal Character Set
UDH	User Data Header
UDP	User Datagram Protocol
UE	User Equipment
UI	User Interface
UID	Unique IDentifier
UMTS	Universal Mobile Telecommunications System
UNI	User Network Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USIM	Universal Subscriber Identity Module
UTC	Coordinated Universal Time
UUID	Universally Unique IDentifier
UX	User Experience
vCard	A format for electronic business cards
VIP	Very Important Person
VoHSPA	Voice over High Speed Packet Access
VoLTE	Voice over Long Term Evolution
VoIP	Voice over IP
W-CDMA	Wideband Code Division Multiple Access
WebRTC	Web Real Time Communication
Wi-Fi	Trademark of Industry Consortium "Wi-Fi Alliance" used as synonym for WLAN (Wireless Local Area Network)

WLAN	Wireless Local Area Network
XCAP	XML Configuration Access Protocol
XDM	XML Document Management
XDMC	XML Document Management Client
XDMS	XML Document Management Server
XML	Extensible Markup Language
XSD	XML Schema Definition
xSIM	Generic reference to different types of SIMs (e.g. USIM, ISIM, etc.)

1.4 Document Cross-References

Ref	Document Number	Title
1	[3GPP TS 23.038]	3GPP TS 23.038 Release 10, 3rd Generation Partnership Project; Alphabets and language-specific information http://www.3gpp.org
2	[3GPP TS 23.040]	3GPP TS 23.040 Release 10, 3rd Generation Partnership Project; Technical realization of the Short Message Service (SMS) http://www.3gpp.org
3	[3GPP TS 23.167]	3GPP TS 23.167 Release 11, 3rd Generation Partnership Project; IP Multimedia Subsystem (IMS) emergency sessions http://www.3gpp.org
4	[3GPP TS 23.221]	3GPP TS 23.221 Release 10, 3rd Generation Partnership Project; Architectural requirements http://www.3gpp.org
5	[3GPP TS 23.228]	3GPP TS 23.228 Release 12, 3rd Generation Partnership Project; IP Multimedia Subsystem (IMS) Stage 2, http://www.3gpp.org
6	[3GPP TS 24.167]	3GPP TS 24.167 Release 10, 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; 3GPP IMS Management Object (MO) http://www.3gpp.org
7	[3GPP TS 24.229]	3GPP TS 24.229 Release 10, 3rd Generation Partnership Project; IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP) http://www.3gpp.org
8	[3GPP TS 24.229-rel11]	3GPP TS 24.229 Release 11, 3rd Generation Partnership Project; IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP) http://www.3gpp.org
9	[3GPP TS 24.229-rel12]	3GPP TS 24.229 Release 12, 3rd Generation Partnership Project; IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP) http://www.3gpp.org

10	[3GPP TS 24.301]	3GPP TS 24.301 Release 11, 3rd Generation Partnership Project; Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3 http://www.3gpp.org
11	[3GPP TS 24.341]	3GPP TS 24.341 Release 10, 3rd Generation Partnership Project; Support of SMS over IP networks; Stage 3 http://www.3gpp.org
12	[3GPP TS 26.114]	3GPP TS 26.114 Release 12, 3rd Generation Partnership Project; IP Multimedia Subsystem (IMS); Multimedia telephony; Media handling and interaction http://www.3gpp.org
13	[3GPP TS 26.141]	3GPP TS 26.141 Release 10, 3rd Generation Partnership Project; IP Multimedia System (IMS) Messaging and Presence; Media formats and codecs http://www.3gpp.org
14	[3GPP TS 33.141]	3GPP TS 33.141 Release 10, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Presence service; Security http://www.3gpp.org
15	[3GPP TS 33.203]	3GPP TS 33.203 Release 10, 3rd Generation Partnership Project; 3G security; Access security for IP-based services http://www.3gpp.org
16	[3GPP TS 33.222]	3GPP TS 33.222 Release 10, 3rd Generation Partnership Project; Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS) http://www.3gpp.org
17	[3GPP TS 33.328]	3GPP TS 33.328 Release 10, 3rd Generation Partnership Project; IP Multimedia Subsystem (IMS) media plane security http://www.3gpp.org
18	[IETF-DRAFT-SIPREC-PROTOCOL]	Session Recording Protocol, Version 02, http://tools.ietf.org/html/draft-ietf-siprec-protocol-02
19	[IETF-DRAFT-Chat]	Multi-party Chat Using the Message Session Relay Protocol, Version 14, March 2, 2012, http://tools.ietf.org/html/draft-ietf-simple-chat-14
20	[IETF-DRAFT-RKEEP]	Indication of support for reverse keep-alive, Version 00, June 21, 2012, http://tools.ietf.org/html/draft-holmberg-sipcore-rkeep-00
21	[IETF-DRAFT-RTCWeb_Overview]	Overview: Real Time Protocols for Brower-based Applications, Version 09, February 14, 2014, http://tools.ietf.org/html/draft-ietf-rtcweb-overview-09
22	[PRD-AA.60]	GSMA PRD AA.60 - “Template Agreement for Interworking” Version 9.0 27 September 2012 http://www.gsma.com/

23	[PRD-IR.33]	GSMA PRD IR.33 - “GPRS Roaming Guidelines” Version 6.0 25 May 2011 http://www.gsma.com/
24	[PRD-IR.58]	GSMA PRD IR.58 - “IMS Profile for Voice over HSPA” Version 4.0 22 November 2012 http://www.gsma.com/
25	[PRD-IR.64]	GSMA PRD IR.64 - “IMS Service Centralization and Continuity Guidelines” Version 7.0, 03 May 2013 http://www.gsma.com/
26	[PRD-IR.65]	GSMA PRD IR.65 - “IMS Roaming and Interworking Guidelines” Version 12.0 14 February 2013 http://www.gsma.com/
27	[PRD-IR.67]	GSMA PRD IR.67 - “DNS/ENUM Guidelines for Service Providers & GRX/IPX Providers” Version 8.0 22 November 2012 http://www.gsma.com/
28	[PRD-IR.74]	GSMA PRD IR.74 - “Video Share Interoperability Specification” 1.4 20 December 2010 http://www.gsma.com/
29	[PRD-IR.79]	GSMA PRD IR.79 - “Image Share Interoperability Specification” 1.4 29 March 2011 http://www.gsma.com/
30	[PRD-IR.84]	GSMA PRD IR.84 - “Video Share Phase 2 Interoperability Specification” 2.2 30 December 2010 http://www.gsma.com/
31	[PRD-IR.88]	GSMA PRD IR.88 - “LTE Roaming Guidelines” 9.0 23 January 2013 http://www.gsma.com/
32	[PRD-IR.90]	GSMA PRD IR.90 - “RCS Interworking Guidelines” v8.0 20 March 2014 http://www.gsma.com/
33	[PRD-IR.92]	GSMA PRD IR.92 - “IMS Profile for Voice and SMS” 7.1 18 September 2013 http://www.gsma.com/
34	[PRD-IR.94]	GSMA PRD IR.94 - “IMS Profile for Conversational Video Service” Version 6.1 23 September 2013 http://www.gsma.com/
35	[RCS51]	GSMA PRD RCC.07 RCS 5.1 - Advanced Communications: Services and Client Specification version 4.0 28 November 2013 http://www.gsma.com/

36	[RCS5-SIMPLEIM-ENDORS]	GSMA PRD RCC.12 RCS 5.2 Endorsement of OMA SIP/SIMPLE IM 2.0, Version 3.0 07 May 2014 http://www.gsma.com/
37	[RCS5-CPM-CONVFUNC-ENDORS]	GSMA PRD RCC.11 RCS 5.2 Endorsement of OMA CPM 2.0 Conversation Functions, Version 3.0 07 May 2014 http://www.gsma.com/
38	[RCS5-CPM-IW-ENDORS]	GSMA PRD RCC.10 RCS 5.2 Endorsement of OMA CPM 2.0 Interworking, Version 3.0 07 May 2014 http://www.gsma.com/
39	[RCS5-3GPP-SMSIW-ENDORS]	GSMA PRD RCC.08 RCS 5.2 Endorsement of 3GPP TS 29.311 Service level Interworking for Messaging Services, Version 3.0 07 May 2014 http://www.gsma.com/
40	[RCS5-CPM-MSGSTOR-ENDORS]	GSMA PRD RCC.09 RCS 5.2 Endorsement of OMA CPM 2.0 Message Storage, Version 4.0 07 May 2014 http://www.gsma.com/
41	[PRIVACY-API]	GSMA Canadian OneAPI Pilot, Privacy Service Developer Guide v1.7 – November 2011 https://canada.oneapi.gsmworld.com
42	[PRD-RCC.53]	Joyn Device API Specification Version 1.0 http://www.gsma.com/
43	[RFC2047]	MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text IETF RFC http://tools.ietf.org/html/rfc2047
44	[RFC2396]	Uniform Resource Identifiers (URI): Generic Syntax IETF RFC http://tools.ietf.org/html/rfc2396
45	[RFC2425]	A MIME Content-Type for Directory Information IETF RFC http://tools.ietf.org/html/rfc2425
46	[RFC2426]	vCard MIME Directory Profile IETF RFC http://tools.ietf.org/html/rfc2426
47	[RFC2595]	Using TLS with IMAP, POP3 and ACAP IETF RFC http://tools.ietf.org/html/rfc2595
48	[RFC2616]	Hypertext Transfer Protocol -- HTTP/1.1 IETF RFC http://tools.ietf.org/html/rfc2616
49	[RFC2617]	HTTP Authentication: Basic and Digest Access Authentication IETF RFC http://tools.ietf.org/html/rfc2617
50	[RFC2822]	Internet Message Format IETF RFC http://tools.ietf.org/html/rfc2822

51	[RFC3261]	SIP (Session Initiation Protocol) IETF RFC http://tools.ietf.org/html/rfc3261
52	[RFC3263]	Session Initiation Protocol (SIP): Locating SIP Servers IETF RFC http://tools.ietf.org/html/rfc3263
53	[RFC3264]	An Offer/Answer Model Session Description Protocol IETF RFC http://tools.ietf.org/html/rfc3264
54	[RFC3326]	The Reason Header Field for the Session Initiation Protocol (SIP) IETF RFC http://tools.ietf.org/html/rfc3326
55	[RFC3329]	Security Mechanism Agreement for the Session Initiation Protocol (SIP) IETF RFC http://tools.ietf.org/html/rfc3329
56	[RFC3428]	Session Initiation Protocol (SIP) Extension for Instant Messaging IETF RFC http://tools.ietf.org/html/rfc3428
57	[RFC3458]	Message Context for Internet Mail IETF RFC http://tools.ietf.org/html/rfc3458
58	[RFC3501]	INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1 IETF RFC http://tools.ietf.org/html/rfc3501
59	[RFC3550]	RTP: A Transport Protocol for Real-Time Applications IETF RFC http://tools.ietf.org/html/rfc3550
60	[RFC3711]	The Secure Real-time Transport Protocol (SRTP) IETF RFC http://tools.ietf.org/html/rfc3711
61	[RFC3840]	Indicating User Agent Capabilities in the Session Initiation Protocol (SIP), IETF RFC http://tools.ietf.org/html/rfc3840
62	[RFC3841]	Caller Preferences for the Session Initiation Protocol (SIP), IETF RFC http://tools.ietf.org/html/rfc3841
63	[RFC3858]	An Extensible Markup Language (XML) Based Format for Watcher Information, IETF RFC http://tools.ietf.org/html/rfc3858
64	[RFC3862]	Common Presence and Instant Messaging (CPIM): Message Format IETF RFC http://tools.ietf.org/html/rfc3862
65	[RFC3863]	Presence Information Data Format (PIDF) IETF RFC http://tools.ietf.org/html/rfc3863
66	[RFC3903]	Session Initiation Protocol (SIP) Extension for Event State Publication IETF RFC http://tools.ietf.org/html/rfc3903
67	[RFC3966]	The tel URI for Telephone Numbers IETF RFC http://tools.ietf.org/html/rfc3966
68	[RFC3986]	Uniform Resource Identifier (URI): Generic Syntax IETF RFC http://tools.ietf.org/html/rfc3986

69	[RFC4028]	The Session Timers in the Session Initiation Protocol (SIP) IETF RFC http://tools.ietf.org/html/rfc4028
70	[RFC4122]	The Universally Unique IDentifier (UUID) URN Namespace IETF RFC http://tools.ietf.org/html/rfc4122
71	[RFC4479]	A Data Model for Presence, IETF RFC http://tools.ietf.org/html/rfc4479
72	[RFC4480]	RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF), IETF RFC http://tools.ietf.org/html/rfc4480
73	[RFC4483]	A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages IETF RFC http://tools.ietf.org/html/rfc4483
74	[RFC4568]	Session Description Protocol (SDP) Security Descriptions for Media Streams, IETF RFC http://tools.ietf.org/html/rfc4568
75	[RFC4572]	Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP), IETF RFC http://tools.ietf.org/html/rfc4572
76	[RFC4575]	A Session Initiation Protocol (SIP) Event Package for Conference State, IETF RFC http://tools.ietf.org/html/rfc4575
77	[RFC4589]	Location Types Registry, IETF RFC http://tools.ietf.org/html/rfc4589
78	[RFC4648]	The Base16, Base32, and Base64 Data Encodings, IETF RFC http://tools.ietf.org/html/rfc4648
79	[RFC4825]	The Extensible Markup Language (XML) Configuration Access Protocol (XCAP) IETF RFC http://tools.ietf.org/html/rfc4825
80	[RFC4867]	RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs IETF RFC http://tools.ietf.org/html/rfc4867
81	[RFC4961]	Symmetric RTP / RTP Control Protocol (RTCP) IETF RFC http://tools.ietf.org/html/rfc4961
82	[RFC4975]	The Message Session Relay Protocol (MSRP) IETF RFC http://tools.ietf.org/html/rfc4975
83	[RFC5104]	Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF) IETF RFC http://tools.ietf.org/html/rfc5104
84	[RFC5196]	Session Initiation Protocol (SIP) User Agent Capability Extension to Presence Information Data Format (PIDF) IETF RFC http://tools.ietf.org/html/rfc5196

85	[RFC5285]	A General Mechanism for RTP Header Extensions IETF RFC http://tools.ietf.org/html/rfc5285
86	[RFC5438]	Instant Message Disposition Notification (IMDN) IETF RFC http://tools.ietf.org/html/rfc5438
87	[RFC5438Errata]	Instant Message Disposition Notification (IMDN) IETF RFC 5438 Errata ID 3013 http://www.rfc-editor.org/errata_search.php?rfc=5438 (see also section C.2)
88	[RFC5491]	GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations IETF RFC http://tools.ietf.org/html/rfc5491
89	[RFC5547]	A Session Description Protocol (SDP) Offer/Answer Mechanism to Enable File Transfer IETF RFC http://tools.ietf.org/html/rfc5547
90	[RFC5626]	Managing Client-Initiated Connections in the Session Initiation Protocol (SIP) IETF RFC http://tools.ietf.org/html/rfc5626
91	[RFC5627]	Using Globally Routable User Agent URIs (GRUUs) in the Session Initiation Protocol (SIP) IETF RFC http://tools.ietf.org/html/rfc5627
92	[RFC6135]	Alternative Connection Model for the Message Session Relay Protocol (MSRP) IETF RFC http://tools.ietf.org/html/rfc6135
93	[RFC6223]	Indication of Support for Keep-Alive IETF RFC http://tools.ietf.org/html/rfc6223
94	[RFC6455]	The WebSocket Protocol IETF RFC http://tools.ietf.org/html/rfc6455
95	[RFC7118]	The WebSocket Protocol as a Transport for the Session Initiation Protocol (SIP) IETF RFC http://tools.ietf.org/html/rfc7118
96	[GML3.1.1]	OpenGIS® Geography Markup Language (GML) Implementation Specification, Version 3.1.1, OGC 03-105r1 http://www.opengeospatial.org/
97	[CAB_TS]	OMA Converged Address Book (CAB) Specification, Approved Version 1.0, 13 November 2012 http://www.openmobilealliance.org
98	[CONNMO]	OMA Standardized Connectivity Management Objects for use with OMA Device Management, Approved Version 1.0 – 07 Nov 2008 http://www.openmobilealliance.org
99	[CONNMOHTTP]	Standardized Connectivity Management Objects HTTP Proxy Parameters for use with OMA Device Management, Approved Version 1.0 – 24 Oct 2008 http://www.openmobilealliance.org

100	[CPM-SYS_DESC]	OMA Converged IP Messaging System Description, Candidate Version 1.0 – 12 Oct 2010 http://www.openmobilealliance.org
101	[OMA-DM]	OMA Device Management, Approved Version 1.2.1 – 17 Jun 2008 http://www.openmobilealliance.org
102	[DMPRO]	OMA Device Management Protocol, Approved Version 1.2.1 – 17 Jun 2008 http://www.openmobilealliance.org
103	[MMSENC]	OMA Multimedia Messaging Service – Encapsulation Protocol, Approved Version 1.3 – 13 Sep 2011 http://www.openmobilealliance.org
104	[MMSMO]	OMA Management Object for MMS, Candidate Version 1.3 – 28 Jan 2008 http://www.openmobilealliance.org
105	[Location_API]	RESTful bindings for Parlay X Web Services – Terminal Location, Candidate Version 1.1 – 11 Jan 2011 http://www.openmobilealliance.org
106	[Presence]	OMA Presence SIMPLE Specification, 1.1, http://www.openmobilealliance.org/
107	[Presence2.0_DDS]	Presence SIMPLE Data Specification, Approved Version 2.0, 29 September 2009 http://www.openmobilealliance.org/
108	[Presence2.1_DDS]	Presence SIMPLE Data Specification, Approved Version 2.1, 02 October 2010 http://www.openmobilealliance.org/
109	[Presence2.0_TS]	Presence SIMPLE Specification, Approved Version 2.0, 10 July 2012 http://www.openmobilealliance.org/
110	[Presence2.0_RLS_TS]	Resource List Server (RLS) Specification, Approved Version 2.0, 10 July 2012 http://www.openmobilealliance.org/
111	[Presence_Content]	Presence Content XDM Specification, Approved Version 2.0, 10 July 2012 http://www.openmobilealliance.org/
112	[PDE_13]	OMA Presence SIMPLE Data Extensions , Approved Version 1.3, 01 November 2011 http://www.openmobilealliance.org/
113	[PRESENCEIG]	Implementation Guidelines for OMA Presence SIMPLE v1.1 Presence http://www.openmobilealliance.org/
114	[PRESENCEMO]	OMA Management Object for Presence SIMPLE, Approved Version 1.0, 25 February 2010 http://www.openmobilealliance.org
115	[PRESENCE2MO]	OMA Management Object for Presence SIMPLE 2.0, Approved Version 2.0, 10 July 2012 http://www.openmobilealliance.org
116	[PresenceXDM]	Presence XDM Specification, Approved Version 1.1 – 27 Jun 2008 http://www.openmobilealliance.org/

117	[REST RCS API]	OMA RCS Profile of RESTful Network APIs, Draft Version 3.0 – 31 Jan 2014 http://member.openmobilealliance.org/ftp/Public_documents/ARCH/RCS-Profile3/Permanent_documents/OMA-TS-REST_NetAPI_RCSProfile_-V3_0-20140131-D.zip NOTE: Reference to be updated with the candidate version once available
118	[REST WEBRTC SIG API]	OMA RESTful Network API for WebRTC Signalling , Candidate Version 1.0 – 11 Feb 2014 http://member.openmobilealliance.org/ftp/Public_documents/ARCH/Permanent_documents/OMA-TS-REST_NetAPI_WebRTCSignaling-V1_0-20140211-C.zip
119	[RLSXDM]	Resource List Server (RLS) XDM Specification Approved Version 1.1 – 27 Jun 2008, http://www.openmobilealliance.org/
120	[SHARED-XDM]	Shared XDM Specification, Approved Version 1.1 – 27 Jun 2008 http://www.openmobilealliance.org/
121	[SUPL]	Secure User Plane Location, Candidate Version 2.0 – 27 May 2011 http://www.openmobilealliance.org/
122	[SUPLMO]	OMA Management Object for SUPL, Candidate Version 2.0 – 27 Jan 2011 http://www.openmobilealliance.org/
123	[XDM1.1_AD]	XML Document Management Architecture, Approved Version 1.1, 27 June 2008 http://www.openmobilealliance.org/
124	[XDM2.0_AD]	XML Document Management Architecture, Candidate Version 2.0, 16 September 2008 http://www.openmobilealliance.org/
125	[XDM1.1_Core]	XML Document Management (XDM) Specification, Approved Version 1.1, 27 June 2008 http://www.openmobilealliance.org/
126	[XDM2.0_Core]	XML Document Management (XDM) Specification, Candidate Version 2.0, 16 September 2008 http://www.openmobilealliance.org/
127	[XDMIG]	Implementation Guidelines for OMA XDM v1.1, http://www.openmobilealliance.org/
128	[XDMMO]	OMA Management Object for XML Document Management 1.1, http://www.openmobilealliance.org
129	[vCard21]	vCard, The Electronic Business Card, A versit Consortium Specification, 18 Sep 1996 http://www.imc.org/pdi/vcard-21.doc
130	[ISO8601]	ISO 8601:2004 Data elements and interchange formats -- Information interchange -- Representation of dates and times, 18 Mar 2008 http://www.iso.org
131	[W3C WebRTC]	WebRTC 1.0: Real-time Communication Between Browsers http://www.w3.org/TR/webrtc/

132	[W3C WS]	The WebSocket API http://www.w3.org/TR/websockets/
133	[W3C XHR]	XMLHttpRequest http://www.w3.org/TR/XMLHttpRequest/

1.5 Differences to previous specifications

RCS 5.2 evolves on the functionality defined for RCS 5.1. The following sub-sections list the major differences.

1.5.1 New features and procedures

- Configuration enhancements
  - Indication of whether application being configured handles SMS.
- Standalone Messaging
  - Message Store support for storage of messages sent or delivered over SMS/MMS (see sections 3.2.4.7)
- 1-to-1 Chat
  - Support for Message Revocation (removal of deferred message from deferred storage to deliver it using other means, see sections 3.3.4.1.10 and 3.3.5.4)
  - Message Store support for storage of messages sent or delivered over SMS/MMS (see sections 3.2.4.7)
- File Transfer enhancements
  - Common File Storage for File Transfer via HTTP (see section 3.5.4.8.6)
- Audio Messaging (see section 3.11)
  - Enables an RCS user to record and send an audio message to any of his RCS contacts.
- Extensions – Service tags
  - Enables an RCS Extension to make use of the RCS infrastructure either through standard RCS services or Extension to Extension services.
    - Extension capabilities (see section 2.6.1)
    - Extension traffic (see section 3.12)
    - Extension registration (see section 2.4.4)
    - Extension control (see section 2.10.4)

1.5.2 Removed features and procedures

None

1.5.3 Modified features and procedures

Area	Section	Differences from RCS 5.0
Configuration procedures	2.3	Evolution of the parameters
Registration	2.4.6 2.4.7	Clarified P-CSCF discovery and handling of connectivity loss
SIP OPTIONS request	2.6.1	Removal of capabilities from Accept-Contact header (except those for in-call services)
Presence Capability document	2.6.1.2	Clarified handling of entity attribute and Contact element
Presence Handling	A.1.1	Improve handling of publish expiry parameter
Capability Exchange	2.6.3 2.6.4.1	Reduce number of capability exchange requests sent to non-RCS contacts Send capability exchange requests only to contacts with MSISDN in specific format
APN	2.9.1	Include place holder for clarification of bearer and APN aspects for non-IMS protocols
Message Store	3.2.6.2 B.4	Clarified Synchronization Procedures for Message Store
Message Store	[RCS5-CPM-MSGSTOR-ENDORS]	Modified storage structure
Message Store	[RCS5-CPM-CONVFUNC-ENDORS]	Store actual IMDNs
Chat	3.3.4.2	Clarified Message Size handling
Group Chat	A.1.3.3	Limit which non-RCS users can be invited for a Group Chat
FT via HTTP	3.5.4	Limit number of upload retries to 3
Video Call over LTE	A.1.14	Introduce parameter to disable IR.94 based video calling on RCS devices

Table 1: Modifications from RCS 5.1

2 RCS 5 General Procedures

2.1 RCS 5 architecture

For RCS, the one mandatory network element is the IMS core system which enables peer-to-peer communication between RCS clients. Other network nodes can be deployed by the Service Provider to provide additional parts of the RCS feature set. Figure 3 illustrates a simplified example of the RCS architecture; a Service Provider may choose a different approach to implement a function within the Service Provider domain not influencing the interoperable NNI aspects.

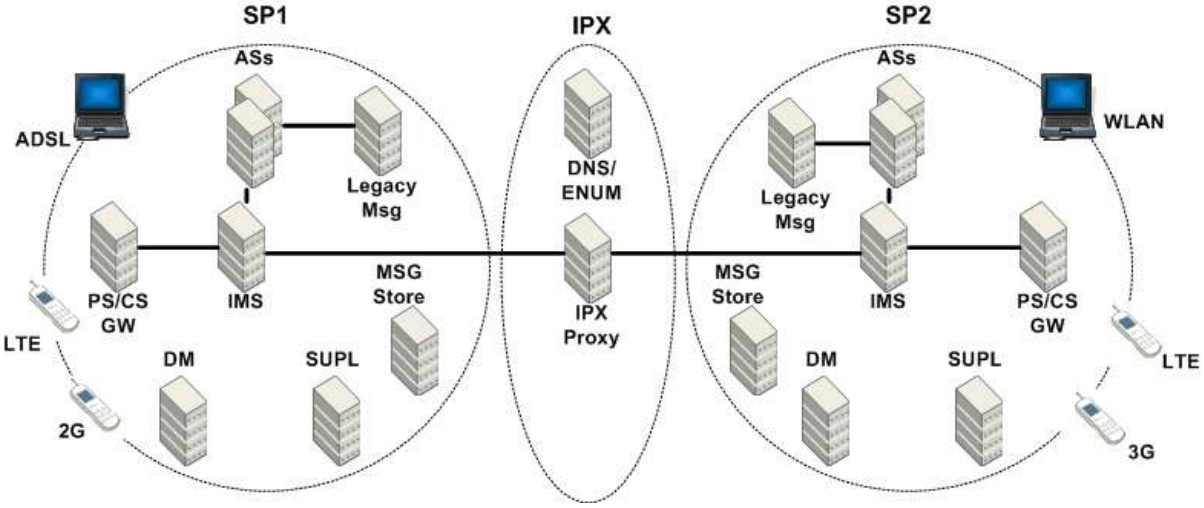


Figure 3: Simplified Example of RCS Architecture

The PS/CS gateway (GW) is used for interworking between Circuit Switched (CS) and Packet Switched (PS) voice, for example Voice over Long Term Evolution (VoLTE). SUPL indicates the Secure User Plane Location element as documented in [SUPL] to support exchanging geolocation information as part of Social Presence Information (SPI) and Geolocation PUSH and PULL. Msg Store relates to the CPM (Converged IP Messaging) Message Store Server as illustrated in section 3.2. Legacy Msg refers to the Short Message Service (SMS)/Multimedia Message Service (MMS) services that may be utilized via an IWF (Interworking Function) located in the group of Application Servers (ASs) which in addition to these IWF node(s) may also include various other nodes used by the RCS services, for example:

- Presence Server
- Messaging Server
- XML (Extensible Markup Language) Document Management (XDM) Server (XDMS)
- Multimedia Telephony (MMTEL) Application Server
- Video Share Application Server, as utilized in [PRD-IR.84]

Figure 3 shows examples of two RCS Service Providers exchanging traffic with each other using the standard Network-to-Network Interface (NNI) mechanisms (IPX, IP Packet Exchange) as documented in [PRD-IR.90].

RCS compliant access networks include, but are not limited to, those illustrated in the Figure 3. Thus, deploying the RCS service does not indicate a 3G network should always be deployed. Further details of RCS services relating to particular access networks are found in section 2.7.

2.2 RCS device modes, client types and device architecture

2.2.1 RCS Device Modes

RCS defines these modes of devices for the telephony service:

- **RCS-VoLTE mode**:
  The mode of operation that an RCS and VoLTE capable (and optionally CS capable) device is using when on LTE (Long Term Evolution) access (e.g., LTE smart phone with VoLTE support that is using VoLTE). A device in RCS-VoLTE mode is configured for VoLTE as described in section 2.9.1 and uses VoLTE to provide the telephony service. When a device capable of VoLTE uses CS to provide telephony service it is in RCS-CS mode (see below). RCS IP Voice/Video Calls are not allowed for devices in RCS-VoLTE mode;
- **RCS-VoHSPA mode**:
  The mode of operation that an RCS and Voice over High Speed Access (VoHSPA) capable (and optionally CS capable) device is using when on High Speed Access (HSPA) (e.g. 3G smart phone using VoHSPA). A device in RCS-VoHSPA mode is configured for VoHSPA as described in section 2.9.1 and uses VoHSPA to provide the telephony service. When a device capable of VoHSPA uses CS to provide telephony service it is in RCS-CS mode (see below). RCS IP Voice/Video Calls are not allowed for devices in RCS-VoHSPA mode;
- **RCS-AA mode**:
  The mode of operation of an access agnostic RCS device using RCS IP Voice/Video Call that cannot provide 3GPP/3GPP2 calls (e.g. a PC notebook with an LTE stick or broadband access network connection, or a tablet);
- **RCS-CS mode**:
  The mode of operation of a device using CS voice as the telephony service (e.g. a 2G/3G/HSPA smart phone without VoHSPA support attached to the CS network for telephony, a CSFB attached RCS smart phone in LTE, and RCS smart phone with CS and Wi-Fi connectivity). RCS IP Voice/Video Calls are possible if available.

  NOTE: If during a transition period a device provides both RCS and VoLTE clients as completely separate implementations, the RCS client shall behave as a non-embedded client and shall consider the device to be in RCS-CS mode whenever in cellular coverage (e.g. the ALWAYS USE IMS APN configuration parameter defined in Table 92 can be used with all values).

For example:

- A device with VoLTE/VoHSPA support which is not configured to use VoLTE/VoHSPA as described in section 2.9.1 (e.g. no subscription supporting these services), or has performed Single Radio Voice Call Continuity (SR-VCC) to 2G/3G, is considered to be a device in RCS-CS mode.
- A device in RCS-CS mode that then turns off radio access may still use Wi-Fi, and become a device in RCS-AA mode.
- A device in RCS-CS mode which has radio access turned on and is configured not to use CS voice call, is considered to be in RCS-AA mode.

The RCS device mode is based on the capability of the device, the access control, and configuration, as per the following table:

Capability of terminal	Access Used	Telephony Service Provided	RCS IP Voice/Video Call Availability and Status	Device Mode
VoLTE, VoHSPA, CS Voice, RCS IP Voice & Video Call	LTE with VoLTE support	VoLTE	Not available	RCS-VoLTE
	LTE without VoLTE support	CS Voice	Available	RCS-CS
	HSPA with VoHSPA support	VoHSPA	Not Available	RCS-VoHSPA
	HSPA without VoHSPA support	CS Voice	Available	RCS-CS
	3G/2G	CS Voice	Available on 3G not on 2G	RCS-CS
	Non-3GPP/3GPP2 access NOTE: not used when device coverage allows VoLTE or VoHSPA	RCS IP Voice & Video Call	Available	RCS-AA
VoLTE, CS Voice, RCS IP Voice & Video Call	LTE with VoLTE support	VoLTE	Not Available	RCS-VoLTE
	LTE without VoLTE support	CS Voice	Available	RCS-CS
	HSPA with and without VoHSPA support	CS Voice	Available	RCS-CS
	3G/2G	CS Voice	Available on 3G not on 2G	RCS-CS
	Non-3GPP/3GPP2 access NOTE: not used when device coverage allows VoLTE	RCS IP Voice & Video Call	Available	RCS-AA
VoHSPA, CS Voice, RCS IP Voice & Video Call	HSPA with VoHSPA support	VoHSPA	Not available	RCS-VoHSPA
	HSPA without VoHSPA support	CS Voice	Available	RCS-CS
	3G/2G	CS Voice	Available on 3G not on 2G	RCS-CS
	Non-3GPP/3GPP2 access NOTE: not used when device coverage allows VoHSPA	RCS IP Voice & Video Call	Available	RCS-AA
CS Voice, RCS IP Voice & Video Call	LTE	CS Voice	Available	RCS-CS
	HSPA	CS Voice	Available	RCS-CS
	3G/2G	CS Voice	Available	RCS-CS
	Non-3GPP /3GPP2 access	RCS IP Voice & Video Call	Available	RCS-AA
RCS IP Voice & Video Call	LTE	RCS IP Voice & Video Call	Available	RCS-AA

	HSPA	RCS IP Voice & Video Call	Available	RCS-AA
	3G/2G	RCS IP Voice & Video Call	Available	RCS-AA
	Non-3GPP/3GPP2 access	RCS IP Voice & Video Call	Available	RCS-AA

Table 2: RCS Device Modes

A device in RCS–AA mode shall take into account the handling defined for RTP and RTCP NAT Traversal defined in section 2.8.1.

Table 3 summarizes the sections in [PRD-IR.92] and [PRD-IR.58] that apply and do not apply to an RCS IP Voice Call or RCS IP Video Call, and where relevant provides a reference to the section where alternative procedures are found.

Document(s)	Relevant sections from [PRD-IR.92], [PRD-IR.58], [PRD-IR.94]	Applicability
[PRD-IR.92], [PRD-IR.58], [PRD-IR.94]	2.2.1 SIP Registration Procedures	The MMTEL IMS Communication Service Identifier (ICSI) and *+g.gsma.rcs.telephony* filled with the values *cs*, *volte*, *vohspa* and/or *none* as appropriate (see section 2.4.3) are placed in the *Contact* header field. In addition, • if RCS IP Voice Call is enabled but IP Video Call is not enabled the client shall also include *+g.gsma.rcs.ipcall* in the *Contact* header field • if RCS IP Video Call is enabled the client shall also include *+g.gsma.rcs.ipcall;video* in the *Contact* header field. • if RCS IP Video Call is enabled and the client supports the specific behaviour when receiving an RCS IP Video Call that cannot be downgraded by the user into an RCS IP Voice Call (see section 3.9.4.1.1) the client shall also include *+g.gsma.rcs.ipcall;+g.gsma.rcs.ipvideocallonly;video* in the *Contact* header field. See also section 2.4, 3.8.4 and 3.9.4
[PRD-IR.92], [PRD-IR.58]	2.2.2 Authentication	Not applicable. See section 2.13
[PRD-IR.92], [PRD-IR.58]	2.2.3 Addressing	All applies except that for RCS, GRUU support in the UE is required. See section 2.4.2

[PRD-IR.92], [PRD-IR.58], [PRD-IR.94]	2.2.4 Call Establishment and Termination (2.2.2 Call Establishment and Termination in [PRD-IR.94])	The client shall include the MMTEL ICSI in the *Contact* and *Accept-Contact* header fields for an RCS IP Voice/Video Call as per [PRD-IR.92] and [PRD-IR.94]. For an RCS IP Voice Call which can be upgraded to an RCS IP Video Call, the client shall include the MMTEL ICSI in both the *Contact* and *Accept-Contact* headers and the video tag in just the Contact header as per [PRD-IR.94]. In addition to the above, • For an RCS IP Voice Call the client shall also include *+g.gsma.rcs.ipcall* in the *Contact* header field and in the *Accept-Contact* header field • For an RCS IP Video Call the client shall also include *+g.gsma.rcs.ipcall;video* in the Contact header field and in the Accept-Contact header field. • For an RCS IP Video Call where video media cannot be removed by the user the client shall also include *+g.gsma.rcs.ipcall*;*+g.gsma.rcs.ipvideo callonly;video* in the *Contact* header field and in the *Accept-Contact* header field. NOTE: If a call is set up that is not required to be an end-to-end IP call, the client just includes the MMTEL ICSI and none of the RCS IP Call feature tags. It is a Service Provider's decision whether or not to break out such a call and may depend on the access network used. Also there is no requirement to bypass SR-VCC [PRD-IR.64] in this case if VoLTE is supported in both terminal and network.
[PRD-IR.94]	2.4.1 Integration of resource management and SIP	If the video media stream is not providing for a sufficient Quality of Service (QoS) level, then the UE may, based on its preferences, modify, reject or terminate the SIP session, according to section 6.1.1 in 3GPP TS 24.229.
[PRD-IR.92], [PRD-IR.58]	2.4.2 Integration of resource management and SIP	Not applicable
[PRD-IR.92], [PRD-IR.58]	2.5 SMS over IP	Not applicable

[PRD-IR.92]	3.2.6 Jitter Buffer Management Considerations	Not applicable
[PRD-IR.92]	3.2.7 Front End Handling	Not applicable
[PRD-IR.92], [PRD-IR.58]	4.1 Robust Header Compression	Not applicable
[PRD-IR.92]	4.2 LTE Radio Capabilities	Not Applicable
[PRD-IR.58]	4.2 HSPA Radio Capabilities	Not Applicable
[PRD-IR.94]	4.2 EPS Bearer Considerations for Video	Not Applicable
[PRD-IR.92], [PRD-IR.58]	4.3 Bearer Management	Not Applicable
[PRD-IR.94]	4.3 LTE Radio Capabilities	Not Applicable
[PRD-IR.94]	4.4 HSPA Radio Capabilities	Not Applicable
[PRD-IR.92], [PRD-IR.58]	5.1 IP Version	Not Applicable
[PRD-IR.92], [PRD-IR.58]	5.2 Emergency Service	Subject to local regulation
[PRD-IR.92], [PRD-IR.58]	5.3 Roaming Considerations	Not Applicable
[PRD-IR.92], [PRD-IR.58], [PRD-IR.94]	Annex A: Complementing IMS with CS (A.1 General, A.2 Domain Selection, A.3 SR-VCC, A.4 IMS Voice service settings management when using CS access, A.5 Emergency Service, A.6 Roaming Considerations, A.7 SMS Support) In [PRD-IR.94] Annex A Complementing IMS with CS (A.1 General, A.2 SR-VCC)	Not Applicable
[PRD-IR.92], [PRD-IR.58], [PRD-IR.94]	All other sections	Applicable

Table 3: IR.92 and IR.58 applicability to RCS IP Voice Call and RCS IP Video Call

2.2.2 RCS Client Types

RCS defines two types of clients:

1. **RCS embedded client**: This is the client that is provided as part of the handset implementation and it is fully integrated with the native applications (address book, gallery/file browser application, calling application, etc.). Consequently, the RCS client shall represent the identity of the device as per section 2.4.2 and [3GPP TS 24.229]; the International Mobile Station Equipment Identity (IMEI) shall be used in sip.instance during registration.
2. **RCS downloadable client**: This is a client that may be preinstalled or that has to be downloaded by the user. However it is not part of the device base software (i.e. it has no access to internal Application Programming Interfaces [APIs] and advanced Operating System [OS] functionality). The level of integration with the native applications is limited

to the possibilities permitted by the corresponding mobile OS or OS platform API. Consequently, the RCS client shall represent the identity of the device as per section 2.4.2, but, the IMEI shall not be used in sip.instance during registration.

2.2.3 Multi-IMS service device architecture

As IMS-based services, like RCS, evolve and increase their penetration among handsets, it becomes necessary to define an architecture where different IMS services can co-exist in a single device. Section C.2 summarizes the available approaches providing an overview of their strengths and weaknesses. The approaches discussed comprise of:

- Multi Stack approach
- IMS device API approach
- IMS device with SIP back-to-back user agent and proxy approach

As explained in section C.2.1, the recommended approach would be an architecture that allows having only a single registration to the IMS per device.

2.3 Configuration Procedures

A user can only initiate the use of RCS services once their client is configured and the corresponding subscriber (uniquely identified by the relevant IMS Unique Resource Identifier [URI]; that is a tel URI and/or a SIP URI) is provisioned by the RCS Service Provider to access the RCS services.

Both processes should be performed automatically (e.g. when a subscriber first turns on their RCS capable device and connects with their Service Provider). This gives the end user the impression that the RCS services are working out of the box and minimises operational impacts to Service Providers.

2.3.1 First-time Start of an RCS capable device

A mobile network offering RCS services to its subscriber base should be able to detect when a user connects to the network for the first time with an RCS capable device. Upon detecting a user connection, two processes are triggered to execute:

1. Service provisioning: the process whereby the relevant configuration is performed on the network elements to make the RCS services available to the user (e.g. provisioning an account on the IMS core and relevant application servers).

   NOTE1: In addition to this auto-provisioning on first usage, the service may be provisioned in advance by the Service Provider.

2. Client configuration: the process whereby the network provides the client with its configuration using one of the mechanisms described in section 2.3.3.

As shown in the Figure 4, an RCS capable device must successfully complete service provisioning and configuration procedures before it can be used. The service provisioning and configuration procedure may be triggered in a variety of different ways, including

- An RCS capable device is powered on: as a result, the network may be able to identify or detect that the user/device pair can use RCS services and, as a consequence, trigger the relevant device configuration procedures described in more detail later in this section.

  NOTE2: The triggering process is network specific and outside the scope of this specification.

- The RCS capable device may also be able to perform a customized bootstrap operation (also named factory bootstrap) to trigger a client-initiated Open Mobile Alliance Device

Management (OMA-DM) session towards an OMA-DM server for client configuration purposes.

An alternative to this automated mechanism could be a manually triggered configuration (e.g. by a menu item or requested by an operator in a store).

2.3.1.1 First-time use scenario

The assumption in this scenario is that User A is entitled to access RCS services (because for example User A's tariff includes the ability to use the RCS services) however User A has never used an RCS-capable device before.

Prior to the first-time registration, it is necessary to provision the user on the network (e.g. by auto-provisioning) and to configure the user's RCS client with the correct settings as described further in this section 2.3. When the provisioning and client configuration is completed, the first-time registration procedure can take place.

RCS first-time registration consists of the following:

1. Register (as described in section 2.4)
2. Establish (i.e. find) a subset among User A's existing contacts (if any) who are also RCS users (as described in section 2.6.2).

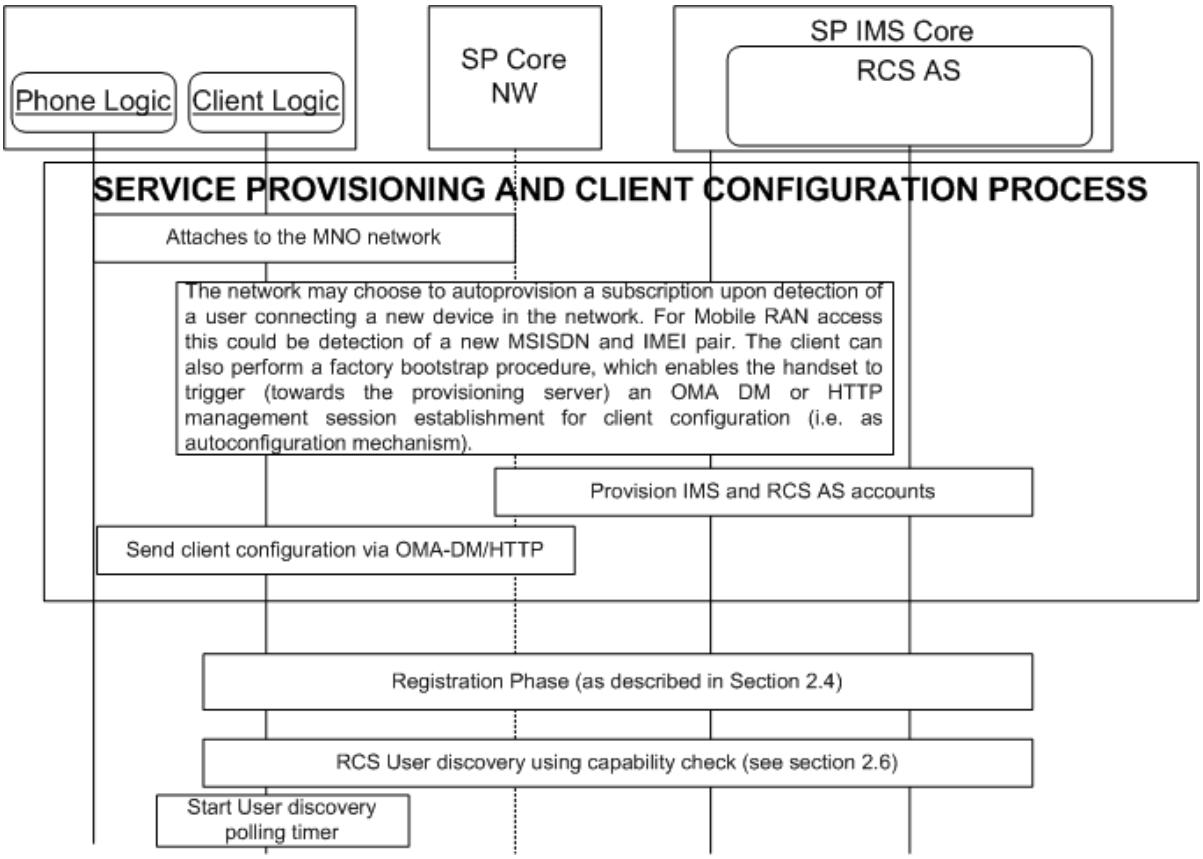


Figure 4: First-Time Start of an RCS capable device - Sequence Diagram

2.3.1.2 Additional first-time configuration scenarios

In addition to the scenario described in the section 2.3.1.1 (first time the user registers with the IMS network), there are several additional scenarios where the same sequence applies:

- When the user changes to another RCS enabled device: In this scenario, the sequence is identical to that described in section 2.3.1.1; however, the IMS provisioning process (i.e. provisioning IMS and RCS AS accounts) is not required as it has already been performed for this subscriber.

- When a customer changes the Subscriber Identity Module (SIM) card in the device: In this scenario, the sequence is identical to the one described in the section 2.3.1.1.
- A configuration update is required that implies changes in the user's IMS identity (that is a different tel URI and/or SIP URI). In this scenario, the sequence is identical to the one described in the section 2.3.1.1.
- A configuration update is required that implies the use of a different capability discovery mechanism: As described in section 2.6, switching the capability discovery mechanism parameter automatically triggers the RCS first-time registration process. This parameter is described in Annex A (section A.1.10).

2.3.2 Client configuration parameters

The set of client configuration settings is presented in Annex A: Managed objects and configuration parameters.

All the RCS client configuration parameters must be restricted from being modified by the user.

In a device supporting RCS-VoLTE or RCS-VoHSPA mode and configured to use those modes when possible (see section 2.2.1), the default IMS settings as defined in [PRD-IR.92]/[PRD-IR.58] are used. Therefore as stated in section A.1.6.1 the client configuration parameters referred to in section A.1.6.2 and those in Table 88 are not used in this case.

Following a successful configuration, the provided settings are active/updated and used on RCS client and the RCS client is ready to register with the network. Once this registration process has successfully completed, the user is able to make use of the RCS services.

These client configuration parameters could also be updated later by the Service Provider by pushing new configuration documents using the OMA-DM enabler or the RCS HTTP configuration mechanisms defined in section 2.3.3.

2.3.3 RCS client autoconfiguration mechanisms

2.3.3.1 Overview

This specification provides two mechanisms that can be used to perform the autoconfiguration of the configuration parameters controlling the RCS functionality in terminals carrying the SIM associated with an RCS user's main identity:

1. [OMA-DM]: This is based on the managed object configuration outlined in Annex A, section A.2. If the RCS capable device supports [OMA-DM], then the following requirements shall be supported by the device:
    - Multiple management authorities in which the Service Provider Device Management accounts are persistent, non-editable and non-visible to an RCS user (e.g. Software (SW) updates do not delete/overwrite DM accounts) and are only accessible to the respective active Service Provider DM account (protected by the OMA-DM Access Control List (ACL) mechanism).
    - The active Service Provider's DM account should be selected and activated after a SIM card change.
    - The RCS configuration parameters are protected against non-Service Provider authorities (through the OMA-DM ACL mechanism).
    - Each RCS Service Provider should have its own RCS Management Object (MO) sub-tree and the OMA-DM account shall have access to the device settings (e.g. for the purpose of access settings configuration).
    - The 'active' Service Provider's RCS MO sub-tree needs to be visible to that Service Provide (i.e. not to the end-user), selected and activated after a SIM card change.

- An RCS capable device shall support the customized bootstrap procedure (also named factory bootstrap (that is the Service Provider OMA-DM account), including OMA-DM server address, is pre-loaded into the device at factory phase). This procedure is specified in section 5.1.2.1 of OMA Device Management Bootstrap (see [OMA-DM]). The customized bootstrap triggers a client-initiated management session from the RCS capable device towards a pre-provisioned OMA-DM server (operated by the Service Provider which the device is subscribing to, i.e. the Home Public Land Mobile Network (HPLMN)). This provides an OMA-DM client with the possibility to initiate and perform the RCS autoconfiguration procedure.
  - The scenarios under which a device shall perform the factory bootstrap procedure[1], are as follows:
    - When a device is switched on for the first time;
    - When a factory reset was performed;
    - When a user changes the SIM card on the device and there is no stored configuration for the new SIM (see section 2.3.4);
    - After a device software update is applied which introduces or extends the RCS client functionality in the terminal
  - After successfully processing the customized bootstrap procedure, the DM client of an RCS capable device shall automatically initiate a management session to the DM server configured in the bootstrap at the next practical opportunity[2] (that is when network connectivity and other factors would allow such a connection to occur).

2. Another RCS client configuration mechanism defined further in section 2.3.3.2 is based on the (Secure) Hypertext Transfer Protocol (HTTP/HTTPS) and has the following main goals:
   - Enabling a configuration procedure that is transparent to the RCS user
   - Reducing the complexity of the auto-detection mechanism on the network infrastructure

For the configuration of additional RCS capable devices (i.e. devices not carrying the SIM associated with a subscriber's main identity), the HTTP(S) mechanism shall be used as described in section 2.3.3.4.

NOTE: Although RCS provides different mechanisms to perform the auto-configuration, the configured parameters remain the same and are independent of the mechanism that is used. The used mechanism only determines the used protocol and the encoding of the parameters between the client and the network.

2.3.3.2 HTTP(S) based client configuration

This mechanism is based on HTTP(S) requests sent by an RCS capable device to an RCS configuration server in order to receive the RCS configuration data.

The HTTP(S) configuration requests may be triggered in two different ways:

[1] The Service Provider network may trigger an OMA-DM configuration when a configured SIM is used in a different device.

[2] This is an additional requirement compared to the OMA DM 1.2.1 specifications.

- **Client-triggered HTTP(S) configuration** if an RCS Service Provider is detected by the client (e.g. SIM-based or by customization).
- **Network-triggered HTTP(S) configuration** if an RCS Service Provider is not detected by the client. It is used to protect pre-installed RCS clients against negative charging impacts in non-RCS networks.

RCS client behaviour is as follows:

- If client-triggered configuration applies: when an RCS capable device boots up (or when the SIM is swapped without rebooting the device [hot swap]) and no valid configuration is available for the used identity, the device sends an initial HTTP request toward the RCS configuration server to verify the current configuration settings' version.
  - If a non-embedded mobile client or a PC client without SIM has no valid configuration for the used identity, this check should be performed each time the RCS client is started.
- After receiving an SMS trigger as described in section 2.3.5, there is an HTTP request sent to the RCS configuration server to verify the current configuration settings' version.
  - If the version available on the client does not match the version on the configuration server, the configuration server will include in its response to the client's HTTP request a configuration document in XML format containing all RCS configuration settings.

  NOTE1: The configuration document is covered in detail in Annex A, sections A.2 and A.3 and based on the OMA Client Provisioning (OMA-CP) syntax.

- In situations where it is necessary to force a reconfiguration of an RCS capable device (e.g. SIM card swap), the device resets the version value of its on-hand RCS configuration settings to 0. The server configuration shall therefore always provide a version value greater than 0.
- In scenarios where the Service Provider disables the RCS functionality on an RCS capable device/client, the HTTP response provided by the RCS configuration server will carry an XML configuration response that carries no configuration parameters and sets the version of the configuration settings to 0 or -1.

The details on the exchanges (e.g. the format employed for each requests) are provided in sections 2.3.3.2.1, 2.3.3.2.2, 2.3.3.2.3, 2.3.3.2.4, 2.3.3.2.5 and 2.3.3.4 of this specification.

This HTTP configuration mechanism operates under the following assumptions:

- As a security measure and to ensure that a Service Provider is able to implement the necessary procedures to resolve a user's Mobile Subscriber Integrated Services Digital Network Number (MSISDN) (that is Remote Authentication Dial In User Service (RADIUS) requests, header enrichment and so on), the configuration of RCS capable devices/clients carrying the SIM associated to an RCS user's main identity can only occur if the device is connected using a mobile PS[3] data network and, therefore, the device should have the necessary Access Point Name (APN) configuration available to perform the connection.

  NOTE2: for other devices/clients the mechanisms defined in section 2.3.3.4 are used.

[3] Please note that if a device does not have a Packet Switched (PS) connection, the autoconfiguration can also happen over Wi-Fi. The decision to implement this mechanism is up to the discretion of each Service Provider.

- As some of the mechanisms presented in the previous paragraphs require an initial HTTP request, an HTTP request is performed first:
  - The device/client shall send an HTTP GET request towards the RCS configuration server's qualified domain name. In this initial HTTP GET request the GET parameters outlined in Table 4 should not be included.
  - As a result of successfully receiving and processing this request, the RCS configuration server returns an HTTP 200 OK response.
  - Upon receiving that HTTP 200 OK response, the RCS client shall perform a second GET request towards the same Uniform Resource Locator (URL) (i.e. the RCS configuration server's qualified domain name) using the HTTPS protocol.
  - The RCS configuration server should be able to correlate both HTTP and HTTPS requests from the same RCS capable device. To achieve this, the configuration server shall provide a cookie as part of the response to the initial HTTP request (Set-Cookie header). The RCS configuration server will expect the client to provide that cookie in the subsequent HTTPS request (in the Cookie header).
- From a UX perspective, the customer is not aware of the auto-configuration process (it is a background process with no pop-ups, alerts or notifications shown to the RCS user on the screen of the RCS capable device) unless the provisioned data includes a message for the end user.

It should also be noted that this mechanism also contributes to reducing the complexity of associated with the auto-detection mechanism as the device will proactively request an update of the configuration settings each time the device is rebooted (or in case of a non-embedded mobile client, on each client start).

2.3.3.2.1 Initial Request

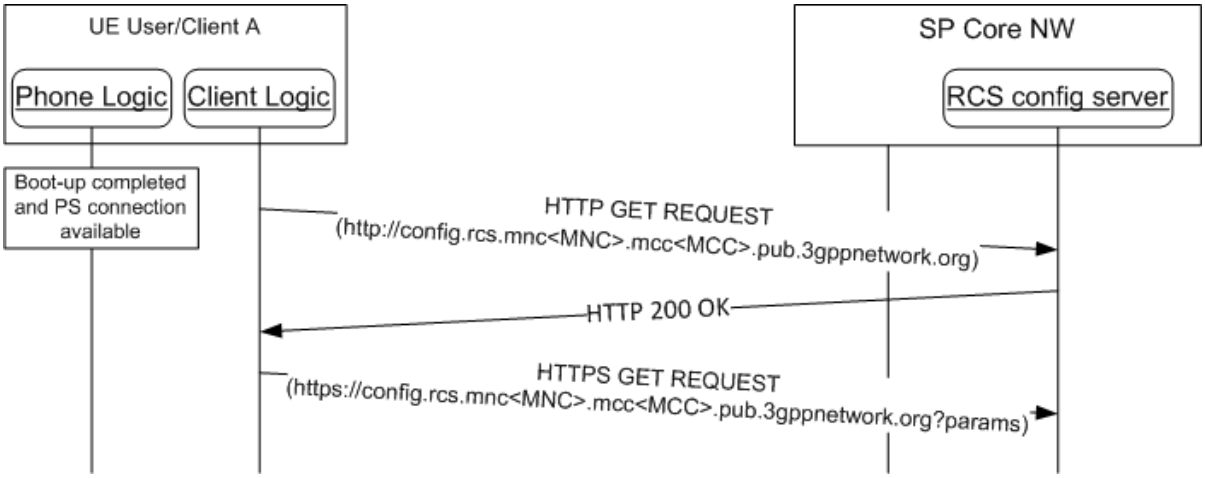


Figure 5: HTTP configuration: Initial requests

Parameters: The following information is included as HTTP GET parameters using a query string:

Parameter	Description	Mandatory	Format
vers	This is either -3, -2, -1, 0 or a positive integer. 0 indicates that the configuration must be updated (e.g. the configuration is damaged; non-existent or an update is needed following a SIM change). A positive value indicates the version of the static parameters (those which	Y	Int (-3, -2, -1, 0 or a positive integer)

	are not user dependent) so the server can evaluate whether an update is required. -1 indicates that the device/client has disabled the RCS services including the autoconfiguration query performed at boot. This may be used by the RCS client/device to inform the SP that the RCS functionality was permanently disabled from the device. -2 Indicates that RCS is disabled on the device (including the configuration query at boot), but a configuration query might be triggered on user action. -3 indicates that RCS is in a dormant state (i.e. no registration) in a way that is transparent to the user. In this state configuration queries are done at the normal trigger points.		
IMSI (International Mobile Subscriber Identity)	If available, the subscriber's IMSI shall be sent as a parameter.	N if the OS platform allows it, it shall be included	String (15 digits)
rcs_version	String that identifies the RCS version supported by the client. It shall be set to "*5.2*" (without the quotes) for clients following this specification.	N, only mandatory from RCS 5.1 onwards	String (4 max), Case-Sensitive
rcs_profile	String that identifies a fixed set of RCS services that are supported by the client. The services that are supported and the value to be used for the rcs_profile parameter to reference to this set are to be defined in external documents (e.g. a Service Provider's RCS Service definition document). In case multiple, (potentially overlapping) sets are supported the parameter shall be included multiple times	N	String (15 max), Case-Sensitive
client_vendor	String that identifies the vendor providing the RCS client.	Y	String (4 max), Case-Sensitive
client_version	String that identifies the RCS client version. client_version_value = Platform "-" VersionMajor "." VersionMinor Platform = Alphanumeric (9 max) VersionMajor = Number (2 char max) VersionMinor = Number (2 char max) Example: client_version=RCSAndrd-1.0	Y	String (15 max), Case-Sensitive

terminal_vendor	String that identifies the terminal OEM.	Y	String (4 max), Case-Sensitive
terminal_model	String that identifies the terminal model.	Y	String (10 max), Case-Sensitive
terminal_sw_version	String that identifies the terminal software version.	Y	String (10 max), Case-Sensitive
IMEI (International Mobile Station Equipment Identity)	If available, the subscriber's IMEI shall be sent as a parameter. Those Service Providers that support a comprehensive device database, they can ignore the terminal_X parameters and use the IMEI instead, if it was available to the RCS implementation.	N if the OS platform allows it, it shall be included	String (15 digits)
friendly_device_name	If provided by the user, a user friendly identification for the device may be passed along that can be used by the network when presenting the user with an overview of their devices.	N, only to be provided if provided by the user	String (30 max before escaping), Case-Sensitive
default_sms_app	This is either 0,1 or 2 0 indicates that the OS does not allow user to select SMS application or the client cannot identify the selected SMS application 1 indicates that the RCS messaging client is selected as the default SMS application 2 indicates that the RCS messaging client is not selected as the default SMS application	N, only mandatory for clients from RCS 5.2 onwards	Int (0,1,2)

Table 4: RCS alternative configuration: HTTPS request GET parameters

Please note that in case of Service Provider-specific RCS clients, the client and terminal vendor, model and version parameters format and values should be agreed with the associated Service Provider prior to any device or RCS client commercialization or update.

- The RCS configuration server URL shall be composed based on the home Service Provider's MCC (Mobile Country Code) and MNC (Mobile Network Code) using a "*config*" subdomain of the domain reserved for RCS services in [PRD-IR.67]. That is: *http://config.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org*
  whereby <MNC> and <MCC> shall be replaced by the respective values of the home network in decimal format and with a 2-digit MNC padded out to 3 digits by inserting a 0 at the beginning (as defined in [PRD-IR.67]).
- The RCS client shall check the MCC and the MNC in the IMSI and compose the configuration Server URL introduced in the previous bullet depending on the HPLMN.
- If an RCS capable device is employed by a Service Provider that does not support RCS, the configuration server URL will not be resolved. In that scenario the application shall handle it as a "client configuration invalid" scenario.

2.3.3.2.2 RCS configuration server response

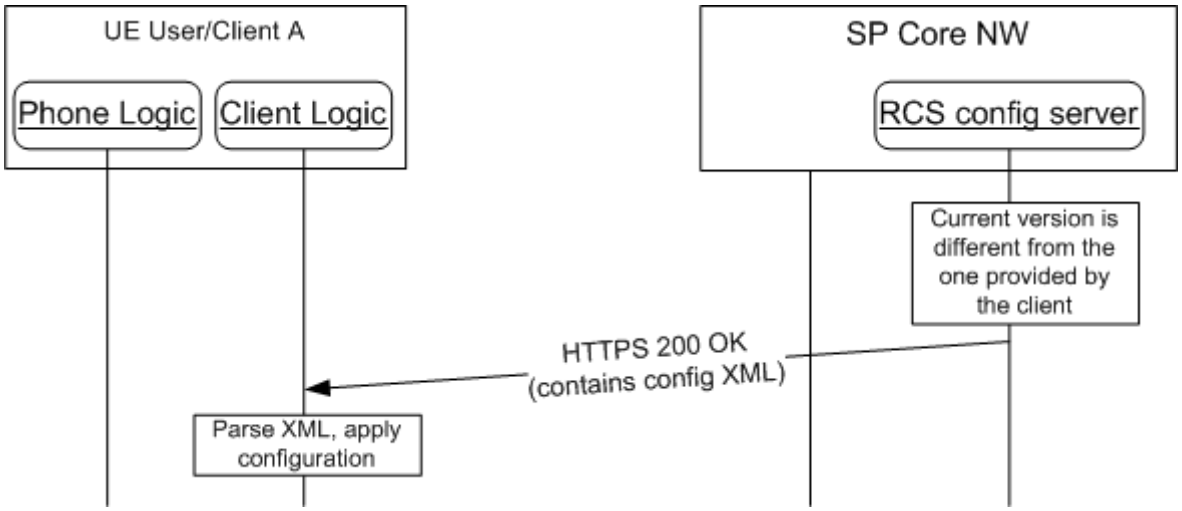


Figure 6: HTTP configuration: Server response

The RCS configuration server (in response to the HTTPS request from an RCS capable device) shall first validate the client and terminal parameters and then checks if the version provided by the client matches the latest version of the configuration available on the server.

The response shall always contain two parameters:

1. The configuration version
2. The validity of the configuration in seconds

If the version matches (i.e. no new configuration settings required), the configuration XML document shall be empty except for the version and the validity parameters:

- The version parameter shall be set to the same value X (as illustrated in Table 5) provided by the client in the HTTPS request
- The validity parameter shall be reset to a server configured value Y (as illustrated in Table 5)

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1">
        <characteristic type="VERS">
                <parm name=”version” value=”X”/>
                <parm name=”validity” value=”Y”/>
        </characteristic>
</wap-provisioningdoc>
```

Table 5: RCS HTTP configuration XML: no configuration changes required

When the RCS client is enabled (i.e. last received configuration had a positive value for version and validity), a change by the user of the selected SMS application that would result in a different value of the default_sms_app parameter shall also trigger a configuration query.

If the Service Provider chooses to temporary disable RCS functionality on the device/client, the response shall carry an XML document containing only the version and validity, both set to 0 as illustrated in Table 6:

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1”>
        <characteristic type="VERS">
                <parm name=”version” value=”0”/>
                <parm name=”validity” value=”0”/>
        </characteristic>
</wap-provisioningdoc>
```

Table 6: RCS HTTP configuration XML: reset RCS client

If the RCS functionality is temporary disabled on an RCS capable device, the device should perform the configuration query each time that it is booted up.

If the Service Provider chooses to permanently disable the RCS functionality on an RCS capable device/client including the configuration query performed at startup, the response shall carry an XML document containing only the version and the validity, both set to -1 as illustrated in Table 7:

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1”>
        <characteristic type="VERS">
                <parm name=”version” value=”-1”/>
                <parm name=”validity” value=”-1”/>
        </characteristic>
</wap-provisioningdoc>
```

Table 7: RCS HTTP configuration XML: reset RCS client and stop configuration query

If the SIM is swapped or the device is reset, the RCS capable device shall again query for configuration settings on each startup assuming that client-triggered HTTP(S) configuration applies. There shall be no other way for the user to trigger a new configuration query. As described in section 2.3.5.1, the RCS client shall also be re-enabled when a SMS message is received requesting a first time configuration.

If the Service Provider chooses to disable the RCS functionality on an RCS capable device/client (including the configuration query performed at startup) until there is a UI dependent user action triggering a new query, the response shall carry an XML document containing only the version and the validity, both set to -2 as illustrated in Table 8:

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1”>
        <characteristic type="VERS">
                <parm name=”version” value=”-2”/>
                <parm name=”validity” value=”-2”/>
        </characteristic>
</wap-provisioningdoc>
```

Table 8: RCS HTTP configuration XML: disable RCS client and stop configuration query

If the SIM is swapped or the device is reset, the RCS capable device shall again query for configuration settings on each startup assuming that client-triggered HTTP(S) configuration applies. As described in section 2.3.5.1, it shall also be re-enabled when a SMS message is received requesting a first time configuration.

If the Service Provider chooses to put the RCS functionality on an RCS capable device/client (including the configuration query performed at startup) in a dormant state, the response shall carry an XML document containing only the version and the validity, with the version set to -3 and a server configured value Y for the validity as illustrated in Table 9:

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1”>
        <characteristic type="VERS">
              <parm name=”version” value=”-3”/>
              <parm name=”validity” value=”Y”/>
        </characteristic>
</wap-provisioningdoc>
```

Table 9: RCS HTTP configuration XML: put RCS client in dormant state

The RCS client shall after receiving such a response behave as follows:

- It shall perform the configuration queries as if it were configured with a valid document (e.g. it performs a query at reboot, when an SMS requesting reconfiguration is received, etc.). In those queries it shall provide as value for the version parameter ‘-3’.
- The existing configuration document remains valid (i.e. a response with version set to ‘-3’ shall be handled in this aspect as if the version parameter matched the current version available in the client).
- It shall not register into the IMS until a subsequent configuration query results in a configuration XML with a positive value. If it was registered when the document with version ‘-3’ is received, the client shall unregister.
- All RCS services entry points shall remain available (including those that base on cached capabilities). When the user activates RCS through one of those capabilities, the client shall:
    - Perform a configuration query providing as value for the version parameter the version of the latest configuration document that was received by the client (i.e. a positive value).
    - If a new configuration document is received or the previously received document is still valid, apply that document, register into the IMS and perform a capability query to verify that the requested action is possible. If not the action is not possible, inform the user of this situation. Keep RCS active afterwards.
    - If an error or a document with a negative version is returned, inform the user that RCS is not available at that time.
    - Provide an indication to the user during these actions to show that RCS is being activated
- These actions to activate RCS shall also be performed when a SMS message requesting reconfiguration is received.

If the server has available an updated RCS configuration for the client, the server response shall contain a configuration XML document (i.e. *Content-Type* of *text/xml*) that the client shall parse and apply:

- The XML format of this document is based on the syntax used in OMA-CP (see Annex A, sections A.2 and A.3 for the details) with a new parameter to include the version, the validity and the message section. A sample of the complete autoconfiguration XML is provided for reference in section A.4.

Server responses that differ from those already described in this section (i.e. an HTTP error) should trigger an RCS capable device/client to try and retrieve RCS configuration settings the next time the device starts (or the RCS client is started). In scenarios whereby the server response consists of a 403 Forbidden error, the device/client implementation shall also remove the current configuration (i.e. as if it had received a response with both validity and version set to 0).

An "RCS Info" MO sub tree shall be included into the RCS management tree containing the configuration parameters described in Table 4, except for "*vers*" and "*IMEI*".

2.3.3.2.3 User Messages

Optionally (that is the tag may not be present), the XML configuration document may be used to convey a user message associated with the result of the configuration server response. The additional XML section is displayed in Table 10:

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1”>
        …
        <characteristic type="MSG">
              <parm name=”title” value=”Example”/>
              <parm name=”message” value=”Hello world”/>
              <parm name=”Accept_btn” value=”1”/>
              <parm name=”Reject_btn” value=”0”/>
        </characteristic>
        …
</wap-provisioningdoc>
```

Table 10: RCS HTTP configuration: User notification/message sample

The meaning of the different parameters is described as follows:

- **Title**: The window title where the user message is displayed.
- **Message**: The message that is displayed to the user. Please note the message may contain references to HTTP addresses (websites) that need to be highlighted and converted into links by the device/client.
- **Accept_btn**: This indicates whether an “Accept” button is shown with the message on the device UI. The action associated with the Accept button on the device/client is to clear the message box.
  A value of 1 indicates that an “Accept” button has to be displayed.
  A value of 0 indicates that no “Accept” button has to be displayed.
- **Reject_btn**: This indicates whether the “Decline” button is shown with the message on the device UI. The action associated with the Reject button on the device/client side is to disable the RCS switch setting in the device.
  A value of 1 indicates that a “Decline” button has to be displayed.
  A value of 0 indicates that no “Decline” button has to be displayed.
  This parameter is optional, when not provided a default value of 0 shall be assumed.

The *MSG* characteristic (i.e. the user message) is optional and will be only present for the following types of RCS configuration server responses:

1. The response containing the full RCS configuration settings.
2. The response disabling RCS configuration on the device (version and validity are set to 0 or a negative value).

The device should display the message and the relevant/configured buttons in the following RCS configuration server response scenarios:

- After receiving the full RCS configuration settings, only if:
  - Working configuration was previously unavailable , including an unavailable working configuration following a SIM change; or
  - Following a terminal reset
- After receiving the disabling RCS configuration response.

The RCS device/client shall send language/locale settings to the server to set the language/locale of the user message. The client should therefore include the HTTP *Accept-Language* header in all the requests and set the value of this header consistent with the device locale.

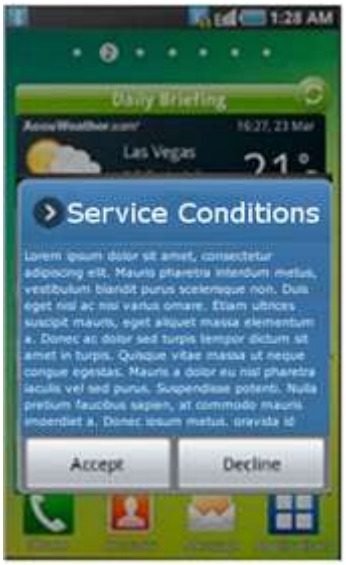


Figure 7 : Autoconfiguration server notification example

2.3.3.2.4 Use Case Overview

Although previously introduced, this section summarizes the different use cases to indicate the corresponding device behaviour for each scenario:

1. **First detection**: This is the first time a user makes use of an RCS device. If the process is successful the device receives the correct configuration XML including the validity period of associated RCS configuration parameters. If the device has no issues (i.e. the device receives no errors) during the registration process, the device refrains from contacting the server again until the validity period has expired. As mentioned previously, this process could require several retries to be attempted until the provisioning in IMS is successfully performed.
   Please note that for those devices not having successfully completed the configuration process yet, any RCS specific UX available on the device should remain disabled (i.e. vanilla behaviour) until a valid RCS configuration is successfully received and processed.
2. **Version checking, no changes**: If the validity period has expired, or the client has been instructed to retry the configuration process, the device sends a request to verify that it has the correct configuration. If the device already has the latest version, the client receives an XML configuration document containing only the same version as the one that was provided by the client already with the validity period reset to a value specified by the RCS configuration server. This indicates that the RCS configuration the device/client currently has is correct and, as a result, the validity period is renewed as indicated by the updated validity parameter value provided as part of the RCS configuration server response.
3. **Version checking, new version available:** If the server has a new version of a subset of the fixed RCS configuration parameters (for example the registration IP address) or if the user has requested a reconfiguration through their Service Provider's Customer Care, the device/client receives a new configuration XML the next time the device/client verifies its version

4. **Validation process is not OK:** If either the RCS device/client or the subscriber is barred from accessing the RCS service, the device will receive an XML with the configuration version and validity attributes set to 0.
   Consequently, the device/client must remove the existing configuration and revert to vanilla behaviour (that is the RCS-specific UX on the device/client is disabled).
5. **SIM change**: If the SIM changes, the previous working configuration should be backed up by the device/client and the device/client should behave as if no configuration is available (that is first-time configuration) and, follow the process described in 2.3.3.2. Please note that if a working configuration backup associated with the new SIM available on the device/client exists, the validity period should be checked and, if it is still valid, the backup working configuration should be used instead of the device issuing a new RCS configuration request.
6. **User with different RCS devices**. If a user uses multiple RCS devices, the same configuration shall be valid for all their RCS devices. The described process shall ensure that the device that the user is currently using has the latest version.
7. **User asks Customer Care to disable (i.e., opt out of) the RCS service**. In this case the user will be un-provisioned from the IMS network, and when the application asks for a reconfiguration it will always receive a XML configuration document with the version and validity set to 0. The RCS service shall remain disabled until the user requests Customer Care to provision their device (i.e. to opt in) for the RCS service again. As a result of disabling the RCS service, the RCS capable device/client shall remove the currently working configuration and disable the RCS-specific UX (that is revert to vanilla behaviour).
8. **User changes selected SMS application.** If the user changes a selected SMS application and client-triggered HTTP(S) configuration applies, this shall trigger a new configuration query with the new value of the default_sms_app parameter.

   NOTE: all scenarios described above comply with one of the following behaviours of the application on the device:

- First time RCS capable device/client utilization: if the RCS capable device/client does not have the correct configuration (version 0 or it is unable to successfully complete the registration process), the device will send a request at each boot sequence (or when the RCS client is restarted) if client-triggered HTTP(S) configuration applies.
- If the RCS configuration server returned a HTTP 511 NETWORK AUTHENTICATION REQUIRED error response on the first time configuration request, the RCS client shall start the SMS based configuration flow as if it were using non-3GPP access (see section 2.3.3.3.1.2).
- The HTTP(S) configuration or re-configuration is triggered as described in section 2.3.5
- If the RCS device/client has received the proper RCS configuration, then it shall not request for a new version unless:
  - The validity period has expired, or,
  - It is not able to complete registration to the IMS

  In these cases, the RCS device/client shall immediately request for a new version and not wait until the next reboot/restart.
- If the response received from the RCS configuration server by the RCS capable client/device is 503 Retry-After, the RCS device/client shall retry the request after the time specified in the “Retry-After” header included as part of the configuration server response.

  - If any other error occurs (for example being unable to resolve the URL or getting an error from the RCS configuration server) the RCS device/client shall retry the procedure during the next time reboot sequence;
  - In the particular case of an RCS client/device receiving a 403 Forbidden, the existing configuration should be removed from the device/client.
- In other error cases (e.g. a 500 Internal Error is issued by the RCS configuration server or the RCS configuration server is unreachable), if a valid configuration is available then, the RCS device/client should keep using it, even if the configuration has expired.
- The following is applicable to both 403 Forbidden and other RCS configuration server error responses:
  - To include scenarios whereby a device migrates to a network without RCS support, the maximum number of unsuccessful consecutive RCS configuration retries allowed by an RCS capable device (including unsuccessful DNS lookup queries) shall be set to 5.
  - If RCS configuration errors persist, the RCS behaviour is disabled at the RCS client/device (that is both the general RCS behaviour if a valid configuration is still available and the RCS configuration sequence performed during the boot sequence).
  - If the SIM is changed or the RCS capable device is reset, the device should again query for RCS configuration settings on every boot sequence if the client-triggered HTTP(S) configuration applies.

Table 11 enumerates all possible RCS configuration server response codes (including error cases):

Response	Use case	Client behaviour
200 OK	Initial HTTP request response	The client sends the HTTPS request including the cookie
503 Retry after	The server is processing the request/provision	Retry after the time specified in the “Retry-After” header
200 OK + XML with full configuration	New configuration sent to the device	Process configuration, try to register and if successful, not try reconfiguration until the validity period is expired, the device/client is restarted or SIM is changed
200 OK + XML with version and validity period only	No update needed	Retry only after validity period, next restart or SIM change
200 OK + XML with version and validity period only and both set to 0	Customer or device are not valid or the customer has been unprovisioned from RCS	Retry only after next restart or SIM change If a configuration was available, it shall be removed from the client.
200 OK + XML with version and validity period only and both set to -1	Customer or device are not valid or the customer has been unprovisioned from RCS	The client shall no longer retry autoconfiguration until SIM is changed or a factory reset performed. If a configuration was available, it shall be removed from the client.

500 Internal Server error (or any other HTTP error except 403)	Internal error during configuration/provisioning	Retry on next reboot/the next time the client starts
403 Forbidden	Invalid request (e.g. missing parameters, wrong format)	The configuration is removed in the device and version is set to 0. Retry on next reboot, the next time the client starts
409 Conflict	A duplicate value was provided for the friendly_device_name	The user should be asked to provide another value for the friendly_device_name parameter and the configuration request should be retried including the new value NOTE: this return code is only applicable to the friendly_device_name as that is the only parameter controlled by the user that could generate a conflict
511 Network Authentication Required	Network-based authentication is not possible (e.g. in case of non-PS access or security enhanced configuration mechanism over PS access).	Client starts non-PS configuration flow as defined in 2.3.3.3 including the cookie if provided.
The RCS configuration server is unreachable	RCS configuration server missing or down	Retry on next reboot, the next time the client starts

Table 11: Summary of RCS autoconfiguration responses and scenarios

2.3.3.2.5 Security considerations

For terminals carrying the SIM associated to the user's main identity the connection is carried out over the PS access network, therefore the current design ensures that it is not possible to perform a man-in-the-middle attack whereby a third party is able to impersonate the RCS configuration server.

To secure interoperability between Service Providers and to reduce complexity on the RCS device/client, the HTTP configuration server shall make use of public root certificates issued by a recognized Certification Authority (CA), that is the root certificates are similar to those used by standard webservers which are widely recognized by browsers and web-runtime implementations both in Personal Computers (PCs) and devices.

To address security concerns due to mobile application system vulnerability (e.g. provisioning of malicious applications that appear to Configuration Server as "trusted" applications), the security enhanced configuration mechanism could be implemented. In that case the procedures to resolve the user's MSISDN (that is RADIUS requests, header enrichment and so on) shall be used only for acquiring the user's MSISDN and not for verifying the user's identity. Specifically, an HTTP 511 NETWORK AUTHENTICATION REQUIRED response shall be generated by the RCS configuration server that contains a cookie as part of the response to the initial HTTP request (Set-Cookie header). The client shall then initiate the SMS based configuration mechanism (see section 2.3.3.3.1.2) without requesting the user to provide its MSISDN. The RCS configuration server shall expect the client to provide that cookie in the subsequent HTTPS request (in the Cookie header).

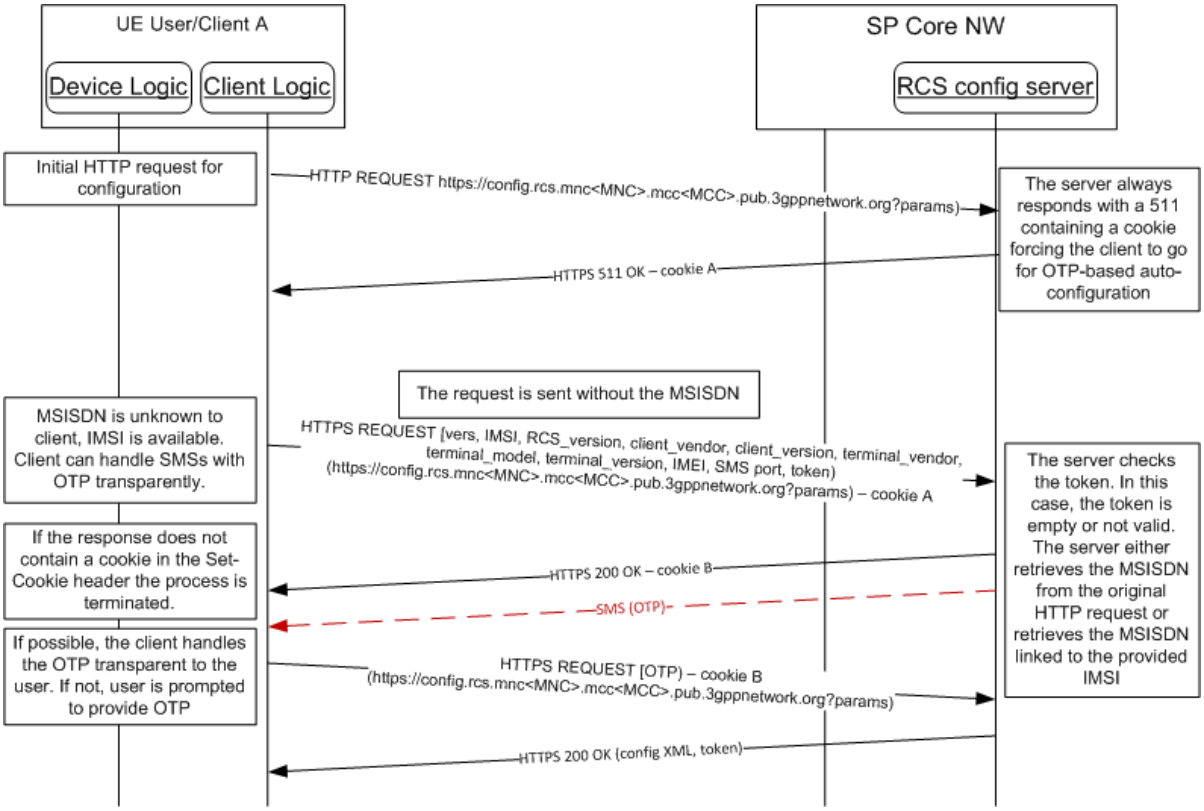


Figure 8: HTTP Configuration: Security enhanced

NOTE: The Service provider shall be able to select between the standard and security enhanced configuration mechanism. It is up to the Service Provider policy to select the most appropriate configuration mechanisms for particular configuration requests.

2.3.3.3 HTTP(S) based client configuration mechanism over non-3GPP access

One of the main limitations of the HTTP configuration mechanism described in section 2.3.3.2 is that it only can take place over PS access as header enrichment is required to identify the RCS subscriber. As an alternative, based on the mechanism presented in section 2.3.3.4 to configure additional devices based on an initial SMS exchange, the current section introduces the process to get a primary device configured when 3GPP PS access is not available to the RCS client.

Finally, note that this mechanism shall only be used when it is not possible to perform the configuration over a PS connection.

2.3.3.3.1 Overview

Depending on the specific solution, the RCS client may be able to identify that it is not possible to perform the RCS configuration over PS access (e.g. because currently only Wi-Fi connections are available). In that case the client can obtain the configuration by following the procedures in section 2.3.3.3.1.2. For clients that are not aware of the connectivity section 2.3.3.3.1.1 provides a specific procedure that can be used for the case where the client can guarantee that any cellular connection in the path to the service provider's configuration server is terminated locally.

2.3.3.3.1.1 Clients not able to identify bearer of configuration request

There is a specific case where the solution is not able to identify whether or not configuration is done over cellular access. In these circumstances a solution that is able to ensure that any cellular connection in the path towards the RCS configuration server is terminated on the device itself (e.g. if Wi-Fi is used it is not tethered to a cellular PS connection), shall perform a first request for configuration using the standard HTTP configuration mechanism described in section 2.3.3.2:

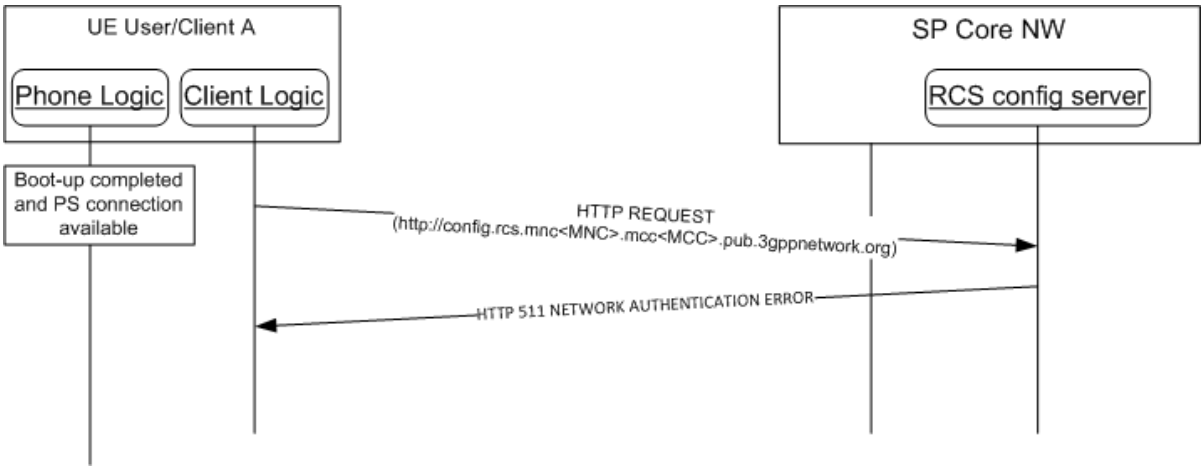


Figure 9: HTTP configuration mechanism: Failed request due to missing header enrichment

NOTE1: The use of another device's PS connection (e.g. a Wi-Fi to cellular-PS-router) may lead to an incorrect identification of the requesting device. Therefore this request can only be sent reliably by clients that can be aware that any PS connection in the path towards the RCS configuration server is provided by themselves.

NOTE2: Most clients connected over Wi-Fi will not be able to verify that there is no cellular connection used further down the path towards the configuration server and should therefore start immediately with a HTTPS request as described in section 2.3.3.3.1.2.

When this initial request is performed over a non-PS access network, the RCS configuration server is unable to successfully identify/verify the identity of the requester (i.e. RADIUS or header enrichment is no longer an option). In this case, the RCS configuration server shall reply with an HTTP 511 NETWORK AUTHENTICATION REQUIRED error response and the client should continue with the procedure described in section 2.3.3.3.1.2. Otherwise the procedure in section 2.3.3.2 shall be followed.

2.3.3.3.1.2 Non-cellular configuration

When performing the configuration over non-cellular access (either because the access is known to be non-cellular or as a result of the procedure in section 2.3.3.3.1.1), the RCS client shall follow the SMS based configuration mechanism as detailed below:

If the MSISDN is unavailable (e.g. a previous RCS configuration procedure has not occurred wherein an RCS client is able to identify the MSISDN because it has been included or as part of the SIP-URI provided in the received XML configuration document) two situations exist:

1. The IMSI is not accessible or the client cannot handle SMS messages with a One-Time Password (OTP) in the background as described in section 2.3.3.3.2. In this case, the

RCS client shall prompt the user to provide a MSISDN (in E.164 format) for the current device unless a valid cookie is provided from a previous RCS configuration server response.
The RCS device performs an HTTPS configuration request in the same manner as described in section 2.3.3.2.1 (Table 4), plus three additional parameters (i.e. MSISDN, SMS_port and token).

2. The IMSI is available and the client can handle SMS messages with the OTP in the background. In this case, the RCS client shall not prompt the user to provide a MSISDN, and perform a HTTPS configuration request with just two additional parameters (i.e. SMS_port and token).

Parameter	Description	Mandatory	Format
vers	This is either -3, -2, -1, 0 or a positive integer. 0 indicates that the configuration must be updated (e.g. configuration is damaged; non-existent or follows a SIM change). A positive value indicates the version of the static parameters (those which are not subscriber dependent) so the server can evaluate whether an update is required. -1 indicates that the device/client has disabled the RCS services including the autoconfiguration query performed at boot. This may be used by the RCS client/device to inform the SP that the RCS functionality was permanently disabled from the device. -2 Indicates that RCS is disabled on the device (including the configuration query at boot), but a configuration query might be triggered on user action. -3 indicates that RCS is in a dormant state (i.e. no registration) in a way that is transparent to the user. In this state configuration queries are done at the normal trigger points.	Y	Int (-3, -2, -1, 0 or a positive integer)
IMSI (International Mobile Subscriber Identity)	If available, the subscriber's IMSI shall be sent as a parameter.	N if the OS platform allows it, it shall be included	String (15 digits)
rcs_version	String that identifies the RCS version supported by the client. It shall be set to "*5.2*" (without the quotes) for clients following this specification.	N, only mandatory from RCS 5.1 onwards	String (4 max), Case-Sensitive

rcs_profile	String that identifies a fixed set of RCS services that are supported by the client. The services that are supported and the value to be used for the rcs_profile parameter to reference to this set are to be defined in external documents (e.g. a Service Provider's RCS Service definition document). In case multiple, (potentially overlapping) sets are supported the parameter shall be included multiple times	N	String (15 max), Case-Sensitive
client_vendor	String that identifies the vendor providing the RCS solution.	Y	String (4 max), Case-Sensitive
client_version	String that identifies the RCS solution version. client_version_value = Platform "-" VersionMajor "." VersionMinor Platform = Alphanumeric (9 max) VersionMajor = Number (2 char max) VersionMinor = Number (2 char max) Example: client_version=RCSAndrd-1.0	Y	String (15 max), Case-Sensitive
terminal_vendor	String that identifies the device OEM.	Y	String (4 max), Case-Sensitive
terminal_model	String that identifies the device model.	Y	String (10 max), Case-Sensitive
terminal_sw_version	String that identifies the device software version.	Y	String (10 max), Case-Sensitive
IMEI	If available, the subscriber's IMEI shall be sent as a parameter. Those Service Providers that support a comprehensive device database can ignore the terminal_X parameters and use the IMEI instead, if it was available to the RCS implementation.	N if the OS platform allows it, it shall be included	String (15 digits)

msisdn	MSISDN in E.164 format of the primary SIM which is used to derive the user's main identity.	N, it is only mandatory if the IMSI is not provided	E.164 (+447900000001) in international format NOTE: In case that msisdn comes with a plus sign, the client shall provide the msisdn value with the plus sign encoded as per [RFC3986] section 2.1.
SMS_port	This parameter sets the UDH port that has to be used for the SMS that is to be employed to validate the requester through a One Time Password (OTP). If set to 0, the client indicates the server that the SMS UDH procedures are not supported either by the client or the platform, so a standard SMS (user visible) shall be used instead. If not set, the default port value used shall be 37273.	N	Int (0-65355)
token	If this is the first time the device is being configured (or the validity of the token is expired), this should be an empty string. If not, the token obtained in the initial configuration process shall be reused.	Y	String
friendly_device_name	If provided by the user, a user friendly identification for the device may be passed along that can be used by the network when presenting the user with an overview of their devices.	N, only to be provided if provided by the user	String (30 max before escaping), Case-Sensitive
default_sms_app	This is either 0,1 or 2 0 indicates that the OS does not allow user to select SMS application or the client cannot identify the selected SMS application 1 indicates that the RCS messaging client is selected as the default SMS application 2 indicates that the RCS messaging client is not selected as the default SMS application	N, only mandatory for clients from RCS 5.2 onwards	Int (0,1,2)

Table 12 : HTTP configuration for primary devices over non-PS access: HTTPS request GET parameters

1. At this point the RCS configuration server is able to identify whether this is a first time request:
   a) If the token value is empty or the provided token is invalid (i.e. first time to configure over a non-PS access network), the request is identified by the RCS configuration server as a first time configuration. In this case, and provided the network allows configuring devices using this mechanism, the RCS configuration server responds with an HTTP 200 OK response that includes a new cookie (Set-Cookie header) to be used in the subsequent HTTP(S) requests.
      i. Following the request, an SMS message shall be sent to the primary RCS capable device, i.e. the device using the SIM associated with the MSISDN or IMSI sent in the HTTP request. This SMS message will contain a one-time password (OTP). The format of this SMS is covered in detail in section 2.3.3.3.2.

   NOTE: the RCS configuration server provider may implement mechanisms on the server to protect it from suspicious or potentially malicious transactions (e.g. a client causing too many SMS)

      ii. In parallel, if OTP handling that is transparent to the user is not possible, the device performing the HTTP configuration prompts the user for the OTP. Therefore, the user should manually enter the value that was received via SMS.

   NOTE: to handle scenarios wherein it is not possible for the network to send an SMS message in the format of section 2.3.3.3.2 to the device, an RCS client should always permit the user to enter the OTP, potentially after some initial delay

      iii. The device performing the HTTP configuration makes a second HTTPS request using the following parameters in the GET request:

Parameter	Description	Mandatory	Format
OTP	This is the password received on the primary device using the SIM associated with the provided MSISDN/IMSI	Y	String

Table 13: RCS HTTP configuration for primary devices: Second and final HTTPS request GET parameters

   NOTE: the second HTTPS request shall include the cookie obtained in step 1 (cookie header) so that the RCS configuration server is able to correlate the initial and subsequent HTTPS requests.

      iv. From this point the procedure is identical to the one described in sections 2.3.3.2.2 and 2.3.3.2.3. If the response includes a full XML configuration document however, the generated token is added as a parameter to the configuration server's 200 OK response.

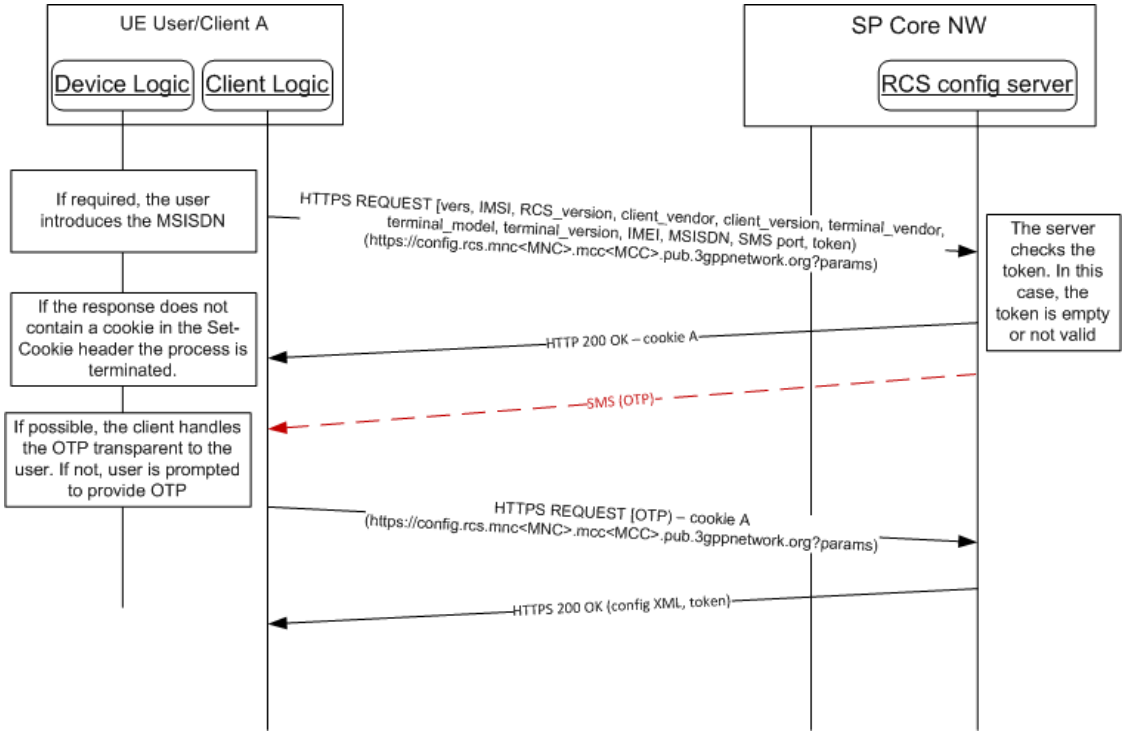


Figure 10: RCS HTTP configuration for primary devices over non-PS access with MSISDN: invalid token

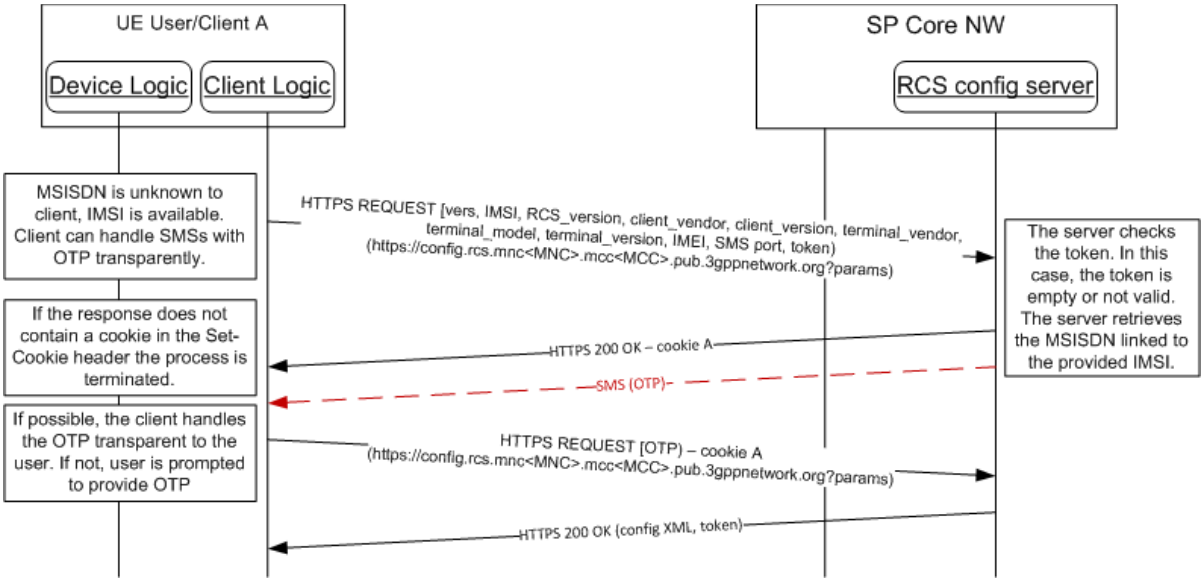


Figure 11: RCS HTTP configuration for primary devices over non-PS access with only IMSI: invalid token

b) If the token is valid (i.e., non-empty, and successfully verified by the configuration server), then, from this point the procedure is identical to the one described in sections 2.3.3.2.2 and 2.3.3.2.3. If the response includes a full XML configuration document however, the generated token is added as a parameter to the configuration server's 200 OK response.

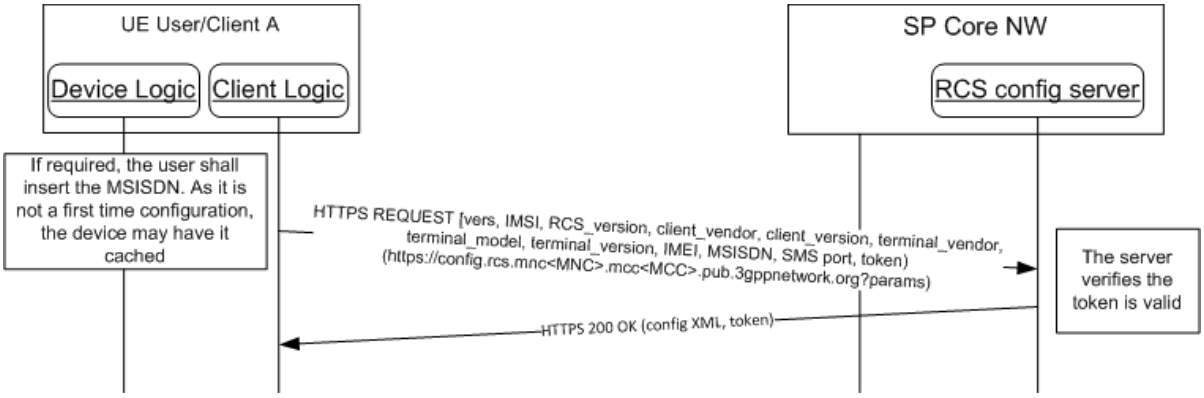


Figure 12: RCS HTTP configuration for primary devices over non-PS access: Valid token

In order to incorporate the token in the RCS configuration server responses, a new characteristic, TOKEN, is provided at the same level at the VERS characteristic so it may be provided in all other kinds of RCS Configuration server responses (e.g. empty configuration XML [with and without a message] and with a full configuration XML):

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1">
        <characteristic type="VERS">
                <parm name=”version” value=”X”/>
                <parm name=”validity” value=”Y”/>
        </characteristic>
        <characteristic type="TOKEN">
                <parm name=”token” value=”Z”/>
                <parm name=”validity” value=”W”/>
        </characteristic>
        -- Rest of the XML if applicable
</wap-provisioningdoc>
```

Table 14: RCS HTTPS configuration XML: Token characteristic

In both cases (i.e. bullets a) and b) above), the token and MSISDN shall be stored on the RCS capable device. Therefore it is not necessary to re-execute the entire procedure for future requests. These values shall be removed together with the rest of the RCS configuration when the device or RCS client is reset.

2.3.3.3.2 SMS format to receive the OTP value

Since in case of a primary device configuration, the device receiving the SMS containing the OTP password shall match the device where the RCS client is running, the preferred approach is that the SMS is sent in a format that allows the client to intercept the OTP in a transparent manner. In order to do so, the RCS configuration server shall perform the following steps:

1. If the value for the SMS_port parameter included in the HTTPS request sent by the device after receiving a HTTP 511 error response is a positive integer in the range between 1 and 65535 and the RCS configuration server supports the UDH (User Data Header) handling procedure (as per [3GPP TS 23.040]) to send a SMS to a specific port, then the following SMS format convention shall be used:
   - DataCodingScheme = 08 (UCS2)
   - UserDataHeader = 06 05 04 4074 0000
     - UDHL length fields=06 05 04,

- Destination port: port provided by the client in HTTP request encoded in hex. If not provided, 37273 (0x9199) shall be the default value.
- Source Port: 0000 (0 in decimal)

- Content of the message shall be the OTP encoded in the same format the Service Provider uses to transmit user readable SMSs.

With this convention, an SMS sent to the device shall be routed to an application listening for SMS on the port indicated by the client and shall be handled transparently to the user.

2. If SMS_port is set to 0, the UDH procedures are not supported either by the client or the platform/OS the client runs on. Consequently, the server shall send a standard SMS and the user shall be prompted by the client to manually provide the OTP code to the RCS client (e.g. via a text box).

For the case that the Service Provider wants to send a standard SMS for the OTP code, the SMS_port parameter shall be included in the HTTPS 200 OK response sent by the RCS configuration server just before the SMS that carries the OTP code (see HTTP flows presented in figures 9 and 10). The response shall carry an XML document containing the SMS_port parameter set to 0 as illustrated in Table 15:

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1">
      <characteristic type="POLICY">
            <parm name="SMS_port" value="0"/>
      </characteristic>
</wap-provisioningdoc>
```

Table 15: RCS HTTPS configuration XML: SMS_port zero policy

In this case the OTP handling that is transparent to the user is not possible and the client prompts the user to enter the OTP which is received via SMS.

NOTE: The Service Provider should allow enough time prior to sending the SMS with the OTP code so as to make sure that the client has received the HTTPS 200 OK response that carries the XML with the SMS_port parameter set to zero. This response shall be sent always considering the rcs_version parameter and thus ensuring backward compatibility. In case that the Service Provider sets a different value to the SMS_port parameter, this value shall be ignored by the client.

2.3.3.3.3 Use cases review

The error conditions and use case scenarios covered in section 2.3.3.2.4 also apply for configuration over non-PS access, but in this case any disabling of the client shall be limited to that specific non-PS network. Further configuration attempts shall thus be done when the device connects to a cellular or another non-PS network. In addition to those errors, for the process of performing a configuration over non-PS access networks the following specific error conditions shall be taken into account and supported:

1. In the scenario whereby a user has to be prompted to provide an MSISDN, the given MSISDN may be invalid or unauthorized to retrieve the RCS configuration. As a result, the initial request shall be answered by the RCS configuration server with an HTTP 403 FORBIDDEN error response. The RCS client shall inform the user of the problem and may offer to retry with a different MSISDN.

NOTE1: if the MSISDN belongs to a SIM which is not currently available to the RCS capable device, the SMS sent by the configuration server will not be received. Therefore in this scenario, the client should provide a timeout mechanism and prompt the user after the timeout period has been reached, to re-enter a MSISDN.

NOTE2: the timeout mechanism utilized by the RCS client is not in scope of this specification.

2. In the scenario where only the IMSI is sent to the network, the network may not support this type of configuration. In that case the RCS Configuration Server shall answer to the initial request with a HTTP 403 FORBIDDEN response. The RCS client shall in this case request the user for their MSISDN and perform the procedure including the MSISDN. This is shown in the flow in Figure 13:

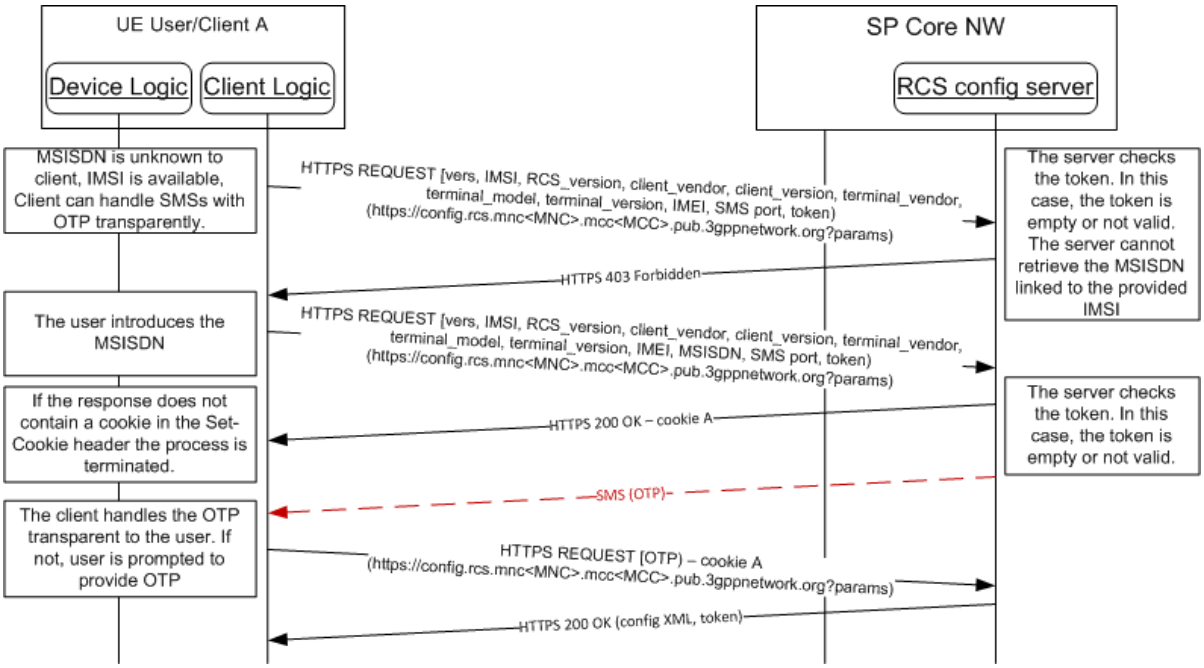


Figure 13: RCS HTTP configuration for primary devices over non-PS access with only IMSI: Not supported by the network

3. The OTP password is invalid. As a result, the RCS configuration server replies with an HTTP 511 NETWORK AUTHENTICATION REQUIRED error response. It is up to the RCS client to provide a user retry mechanism. When retrying, the RCS client shall re-start the configuration process from the beginning.
4. The token is invalid. As a result, the RCS configuration server replies with an HTTP 511 NETWORK AUTHENTICATION REQUIRED error response. It is up to the RCS client to provide a user retry mechanism. When retrying, the RCS client shall re-start the configuration process from the beginning. Consequently, if a valid token was previously stored, it shall be removed from the device.

2.3.3.3.4 Security considerations

The same access security considerations described in section 2.3.3.2.5 for the standard HTTP(S) configuration mechanism also apply in this case.

Service Providers may request the client to fall back to the client configuration mechanism over non-3GPP access while requesting configuration in 3GPP access to secure the user identification via header enrichment, as defined in section 2.3.3.2.5.

In addition, as a Service Provider Option, the RCS configuration server is able to enforce a policy for the OTP challenge on primary devices being always visible to the user, especially for the case where the client would be able to apply it transparent by use of the SMS UDH procedure. If the client receives a Configuration Response in HTTP 200 OK with a "SMS port zero" policy (see section 2.3.3.3.2) then it shall expect the reception of the OTP via user visible SMS. Thus it shall prompt the user to enter the OTP and continue processing with the user input only.

2.3.3.4 Configuration of additional devices sharing the same identity

This section describes the process of autoconfiguration authentication for the scenario in which the SIM associated with the IMS identity is not inserted in the device being provisioned.

2.3.3.4.1 First-time configuration

During first-time configuration, the RCS capable device implementation/client will receive the credentials associated with the primary SIM card of the user regardless of the type of connection they are using (e.g. Wi-Fi, PS) to reach the RCS Configuration server.

The process is as follows:

1. As an option, the RCS capable device implementation/client will offer the possibility to the user to perform manual provisioning
2. The user is prompted for the MSISDN or SIP URI of the primary device and the Service Provider associated with the primary SIM. The account created is always associated with this primary identity that the user has to input into the application. Please note that, as a pre-condition, the aforementioned identity must already be provisioned using the mechanism described in previous sections.
3. The device performs the HTTPS configuration as presented in section 2.3.3.2.1, however, using the following GET parameters instead of the default ones:

Parameter	Description	Mandatory	Format
vers	This is either -3, -2, -1, 0 or a positive integer. 0 indicates that the configuration must be updated (e.g. configuration is damaged or non-existent). A positive value indicates the version of the static parameters (those which are not subscriber dependent) so the server can evaluate whether an update is required. -1 indicates that the client has disabled the RCS services including the autoconfiguration query performed at boot. This may be used by the RCS client/device to inform the SP that the RCS functionality was permanently disabled from the device. -2 Indicates that RCS is disabled on the device (including the configuration query at boot), but a configuration query might be triggered on user action. -3 Indicates that RCS is in a dormant state (i.e. no registration) in a way that is transparent to the user. In this state configuration queries are done at the normal trigger points.	Y	Int (-3, -2, -1, 0 or a positive integer)
msisdn	MSISDN in E.164 format of the primary SIM which is used to derive the identity.	N, Mandatory if sip_uri not provided	E.164 (+447900000001) in international format NOTE: In case that msisdn comes with a plus sign, the client shall provide the msisdn value with the plus sign encoded as per [RFC3986] section 2.1.
sip_uri	SIP URI of the primary device	N, Mandatory if msisdn not provided	String (50 max), Case-insensitive

rcs_version	String that identifies the RCS version supported by the client. It shall be set to "*5.2*" (without the quotes) for clients following this specification.	N, only mandatory from RCS 5.1 onwards	String (4 max), Case-Sensitive
rcs_profile	String that identifies a fixed set of RCS services that are supported by the client. The services that are supported and the value to be used for the rcs_profile parameter to reference to this set are to be defined in external documents (e.g. a Service Provider's RCS Service definition document). In case multiple, (potentially overlapping) sets are supported the parameter shall be included multiple times	N	String (15 max), Case-Sensitive
token	If this is the first time the additional device is being configured (or the validity of the token is expired), this should be an empty string. If not, the token obtained in the initial configuration process shall be reused here.	Y	String (24 max), Case-Sensitive
client_vendor	String that identifies the vendor providing the RCS solution.	Y	String (4 max), Case-Sensitive
client_version	String that identifies the RCS solution version. client_version_value = Platform "-" VersionMajor "." VersionMinor Platform = Alphanumeric (9 max) VersionMajor = Number (2 char max) VersionMinor = Number (2 char max) Example: client_version=RCSAndrd-1.0	Y	String (15 max), Case-Sensitive
device_type	This indicates the type of device where the client is running.	Y	Possible values: - Tablet - PC - Other
friendly_device_name	If provided by the user, a user friendly identification for the device may be passed along that can be used by the network when presenting the user with an overview of their devices.	N, only to be provided if provided by the user	String (30 max before escaping), Case-Sensitive

Table 16: RCS alternative configuration for additional devices: Initial HTTPS request GET parameters

Please note that the initial HTTP request is not required in this case since the header enrichment requirement is not applicable. Therefore, the device implementation/client will directly perform the HTTPS request as presented in Figure 14.

4. As this is a first time request, the token value is empty; the request is then identified as a first time configuration. In this case, and provided the network allows for configuring additional devices using this mechanism, the HTTP server responds with a HTTP 200 OK response carrying a new cookie (Set-Cookie header) to be used in the subsequent HTTP requests
   a) Following the request, an SMS message shall be sent to the primary device, i.e. the phone carrying the SIM associated to the MSISDN the user introduced in step 2. This SMS message will contain an OTP. This message shall be a standard SMS (i.e. no UDH procedures required).
   b) In parallel, the device performing the HTTP configuration prompts for the OTP. Therefore, the user should manually introduce the code delivered via SMS to the primary device.
   c) Once the user enters the OTP, the device performing the HTTP configuration makes a second HTTPS request using the following parameters in the GET request:

Parameter	Description	Mandatory	Format
OTP	This is the password received on the device carrying the SIM associated with the MSISDN introduced in step 2	Y	String (8 Max), Case-Sensitive

Table 17: RCS alternative configuration for additional devices: Second and final HTTPS request GET parameters

Please note this second HTTPS request shall carry the cookie obtained in step 4 (cookie header) therefore the HTTP configuration server can correlate the initial and final HTTPS requests.

   d) From this point onwards the procedure is identical to the one described in sections 2.3.3.2.2 and 2.3.3.2.3, however, with token is added as a parameter. If the request is successful, one of the possible 200 OK responses described in section 2.3.3.2.2 is provided. If receiving a full XML and provided the network uses the sip.instance approach for multidevice handling as described in section 2.11, the response shall include the uuid_Value parameters (see Annex A sections A.1.13 and A.2.10 for further reference).

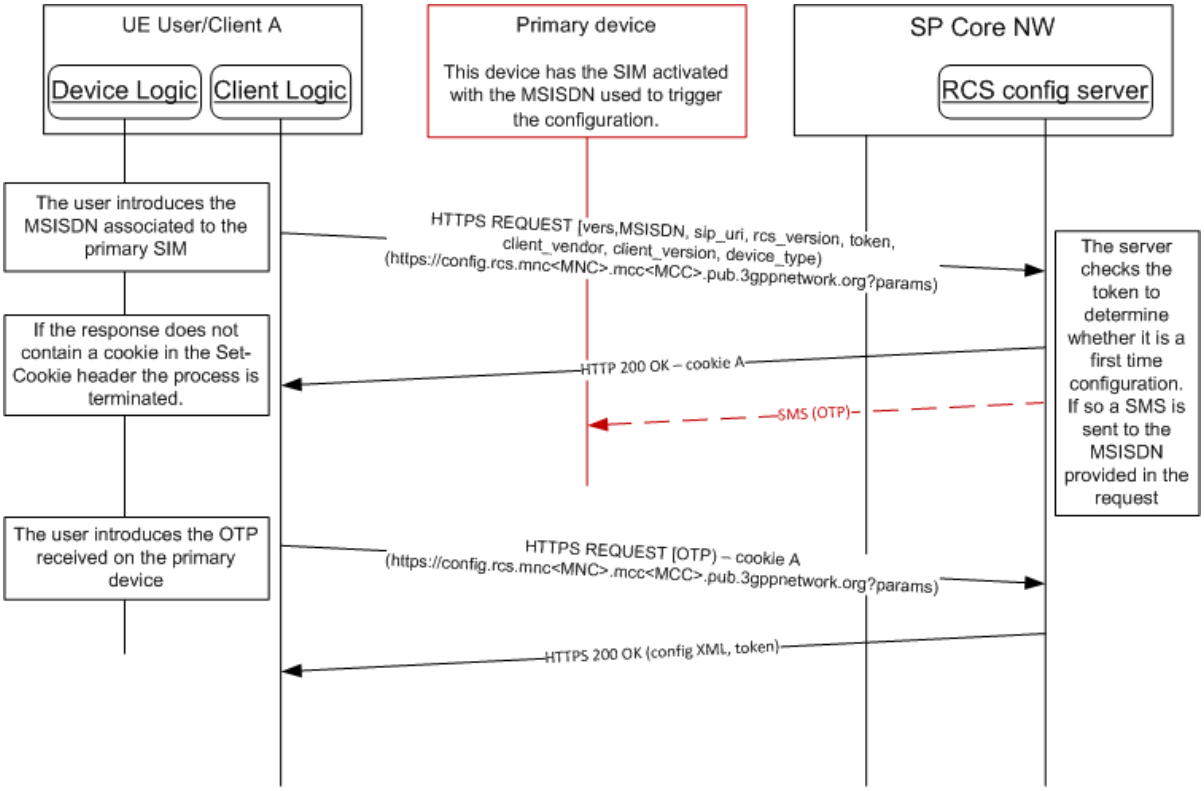


Figure 14: RCS alternative configuration for additional devices: First time configuration

Please note the token shall be stored with the MSISDN so it is not necessary to repeat this procedure for future requests. These values shall be removed together with the rest of the RCS configuration when the device or RCS client are reset.

2.3.3.4.1.1 Error handling

In the process of performing a first time configuration for additional devices, there are three possible error conditions that the client has to be aware of and handle:

1. The MSISDN used is not valid or it is not authorized (including the case the primary MSISDN is not been provisioned yet to use RCS) to get the configuration/make use of the RCS services. In this case, the initial request will be answered with an HTTP 403 FORBIDDEN error and the client shall inform the user of the issue and may offer to retry with a different MSISDN.
2. The OTP password introduced by the user is not valid. In this case, the HTTP configuration server replies again with a HTTP 511 NETWORK AUTHENTICATION REQUIRED error. It is up to the client implementation to offer the user to retry. If retrying, the client shall start the first time configuration process from the beginning.
3. The HTTP server suffers an internal error (HTTP 5XX [except 511], response coming from the server). This case, the user shall be informed of the circumstance and offered to retry. If retrying, the client shall start the first time configuration process from the beginning.

2.3.3.4.2 Subsequent configuration attempts and life cycle

If the client has access to the token and the MSISDN used for the first time configuration, has a value, the initial request is performed

1. An initial request like in the case of the first time configuration of additional devices is made, this time including the token parameter set to the value received on the previous successful configuration attempt

2. If successful, from this point onwards the procedure is identical to the one described in sections 2.3.3.2.2 and 2.3.3.2.3, however, with the token added as a parameter. If the request is successful, one of the possible 200 OK responses described in section 2.3.3.2.2 is provided. If receiving a full XML and provided the network uses the sip.instance approach for multidevice handling as described in section 2.11, the response shall include the uuid_Value parameters (see Annex A sections A.1.13 and A.2.10 for further reference).

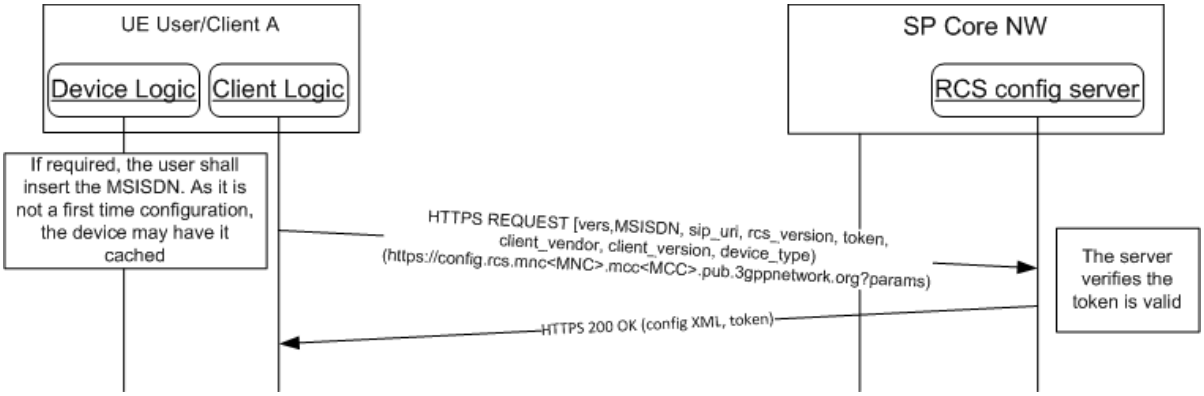


Figure 15: RCS alternative configuration for additional devices: Subsequent attempts

If the token and/or the MSISDN are not available (for example the device is reset), then the client shall start a first time configuration as described in section 2.3.3.4.1.

Please note the received token shall be stored with the MSISDN so it is not necessary to repeat this procedure for future requests. These values shall be removed together with the rest of the RCS configuration when the device or RCS client are reset.

2.3.3.4.2.1 Error handling

In the process of performing a subsequent configuration for additional devices, there are three possible error conditions that the client has to be aware and handle:

1. The MSISDN used is not valid or it is not authorized to get the configuration/make use of the RCS services. In this case, the initial request will be answered with an HTTP 403 FORBIDDEN error and the client shall inform the user of the issue and may offer to retry with a different MSISDN.
2. The token is no longer valid. In this case, the HTTP configuration server replies again with a HTTP 511 NETWORK AUTHENTICATION REQUIRED error. From this moment, the process is equivalent to the first time configuration process after the same error is received.
3. The HTTP server suffers an internal error (HTTP 5XX response coming from the server). This case, the user shall be informed of the circumstance and offered to retry. If retrying, the client shall start the subsequent configuration attempt procedure from the beginning.

2.3.3.4.2.2 Using End User Confirmation Request alternative

As an alternative to the use of SMS to confirm the identity of the user the Service Provider could choose to use the End User Confirmation Request (EUCR, see section 2.10).

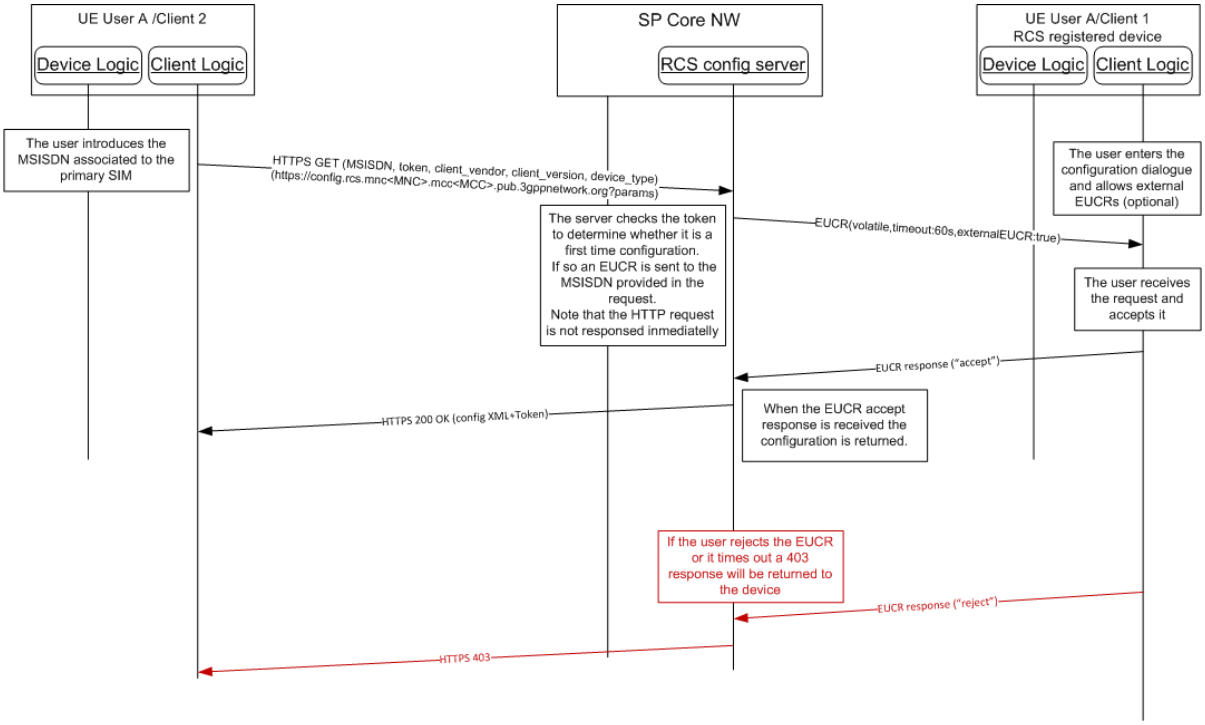


Figure 16: RCS alternative configuration for additional devices: First time configuration using EUCR

The process is very similar to the one described for SMS:

1. As an option, the device implementation/client will offer the possibility to the user to perform manual provisioning as in the SMS mechanism.
2. The user is prompted for the MSISDN or SIP URI of the primary device and the Service Provider associated with the primary SIM as in the SMS mechanism.
3. The device performs the HTTP configuration using the same GET parameters as in the SMS mechanism.
4. At this point the HTTP configuration server is able to identify whether this is a first time request:
   a) If the token value is empty, then the request is identified as a first time configuration.
   b) If the token has a value, it is checked against the HTTP server database. If successful, from this point the procedure is identical to the one described in sections 2.3.3.2.2 and 2.3.3.2.3.

   NOTE1: There is no further authentication of the additional device or the user that starts the configuration process. Appropriate security measures to prevent malicious usage should be implemented on the configuration server.

5. An End User Confirmation Request flow starts for a first-time registration
   a) In case of malicious usage by other person via Internet, the End User Confirmation Request method may block the UI (by unwanted End User Confirmation Request popups) on the device registered for RCS. Therefore, the following optional user dialogue is recommended.
   If implemented on the RCS-registered mobile device, the user enters a UI dialogue to start the configuration of additional RCS devices. This dialogue sets the mobile device into a mode that allows End User Confirmation Requests initiated from an external source. This external source is in this case the user's additional device to configure.

NOTE2: If activation and de-activation of that mode is not implemented on the mobile device, all End User Confirmation Requests are allowed and shown to the user and therefore also the End User Confirmation Requests related to the configuration of the device.

b) A volatile End User Confirmation Request is sent to the MSISDN or SIP URI provided in the HTTP request. The End User Confirmation Request includes the attribute *externalEUCR* set to true.

NOTE3: The HTTP request is not answered immediately.

6. The End User Confirmation Request is received by the device and: will be shown in the user RCS device and:
   a) If the device does allow external End User Confirmation Requests, it will be shown on the user's RCS device. The user may accept it, in which case a 200 OK response is sent as described in section 2.2.3.2.2. Please note that if receiving a full XML and provide the network uses the sip.instance approach for multidevice as described in section 2.11, the response shall include the uuid_Value parameters (see Annex A sections A.1.13 and A.2.10 for further reference). The response will also contain a token to be used in subsequent and future requests:

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1">
        <characteristic type="TOKEN">
                <parm name=”token” value=”X”/>
                <parm name=”validity” value=”Y”/>
        </characteristic>
        <characteristic type="VERS">
                <parm name=”version” value=”Z”/>
                <parm name=”validity” value=”W”/>
        </characteristic>
        <characteristic type="APPLICATION">
                ….
        </characteristic>
</wap-provisioningdoc>
```

Table 18: RCS HTTPS configuration of additional devices using EUCR: First time response to the HTTPS request.

   b) If the device does allow external End User Confirmation Requests but the user rejects the End User Confirmation Request or the timer expires, the server will reply with an HTTP 403 response and the process is concluded (the device is not configured as an end result).
   c) If the device does not allow external End User Confirmation Requests, it shall ignore the request or reject it. As in b) the server will reply with an HTTP 403 response and the additional device is not configured.

NOTE4: there is no further authentication of the additional device or the user that starts the configuration process (i.e. the initial HTTPS request). If misused via Internet, End User Confirmation Request may block an RCS user's UI (by unwanted popups) on the device associated with the primary SIM. Therefore, appropriate security measures to prevent such malicious usage should be implemented.

2.3.3.4.2.3 Using End User Confirmation request with PIN alternative

The Service Provider can add an extra layer of security by using the pin request feature in the End User Confirmation Request.

Using this alternative, the flow is similar as the SMS process described in section 2.3.3.4, except that instead of sending the One-Time Password in the SMS, the One-Time Password is chosen by the user and typed into both devices:

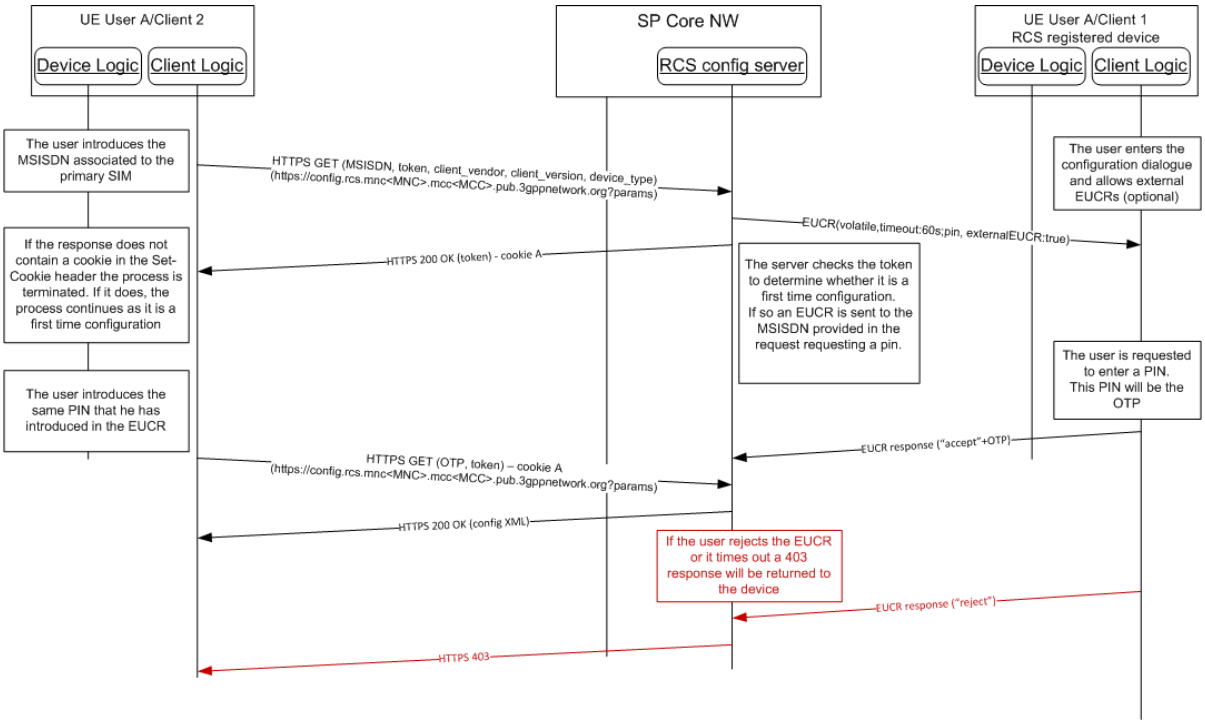


Figure 17: RCS alternative configuration for additional devices: First time configuration EUCR with PIN

NOTE: there is no further authentication of the additional device or the user that starts the configuration process (i.e. the initial HTTPS request). If misused via Internet, End User Confirmation Request may block an RCS user's UI (by unwanted popups) on the device associated with the primary SIM. Therefore, appropriate security measures to prevent such malicious usage should be implemented.

2.3.3.4.2.4 Use cases review

From the use cases presented in section 2.3.3.2.4, only the following scenarios apply to the configuration of additional devices sharing the same identity:

1. First detection
2. Version checking
3. Validation process is not OK
4. User asks Customer Care to disable the RCS service

2.3.3.4.3 Security considerations

The same security considerations described in section 2.3.3.2.5 for the standard HTTP(S) configuration mechanism also apply in this case.

2.3.3.5 Configuration of non-Cellular devices with a dedicated identity

To configure clients on devices that do not carry a SIM, but have to function with a dedicated own identity following generic solution is provided:

1. The user obtains an OTP through means that are out of the scope of this specification (e.g. from an operator website after authentication, delivered together with the device, obtained through an operator's main street shop, etc.)
2. The user is prompted for the E.164 address or SIP URI to be used by the device and their Service Provider. The account created is always associated with this primary identity that the user has to input into the application.
3. The device performs the HTTPS configuration as presented in section 2.3.3.2.1, however, using the following GET parameters instead of the default ones:

Parameter	Description	Mandatory	Format
vers	This is either -3 -2, -1, 0 or a positive integer. 0 indicates that the configuration must be updated (e.g. configuration is damaged or non-existent). A positive value indicates the version of the static parameters (those which are not subscriber dependent) so the server can evaluate whether an update is required. -1 indicates that the client has disabled the RCS services including the autoconfiguration query performed at boot and is therefore never included in the HTTPS GET request. This may be used by the RCS client/device to inform the SP that the RCS functionality was permanently disabled from the device. -2 Indicates that RCS is disabled on the device (including the configuration query at boot), but a configuration query might be triggered on user action. -3 indicates that RCS is in a dormant state (i.e. no registration) in a way that is transparent to the user. In this state configuration queries are done at the normal trigger points.	Y	Int (-3, -2, -1, 0 or a positive integer)

msisdn	E.164 format of the provided identity	N, Mandatory if sip_uri not provided	E.164 (+447900000001) in international format NOTE: In case that msisdn comes with a plus sign, the client shall provide the msisdn value with the plus sign encoded as per [RFC3986] section 2.1.
sip_uri	SIP URI of the device	N, Mandatory if msisdn not provided	String (50 max), Case-insensitive
rcs_version	String that identifies the RCS version supported by the client. It shall be set to "*5.2*" (without the quotes) for clients following this specification.	N, only mandatory from RCS 5.1 onwards	String (4 max), Case-Sensitive
rcs_profile	String that identifies a fixed set of RCS services that are supported by the client. The services that are supported and the value to be used for the rcs_profile parameter to reference to this set are to be defined in external documents (e.g. a Service Provider's RCS Service definition document). In case multiple, (potentially overlapping) sets are supported the parameter shall be included multiple times	N	String (15 max), Case-Sensitive
token	If this is the first time the primary device is being configured (or the validity of the token is expired), this should be an empty string. If not, the token obtained in the initial configuration process shall be reused here.	Y	String (24 max), Case-Sensitive
client_vendor	String that identifies the vendor providing the RCS client.	Y	String (4 max), Case-Sensitive

client_version	String that identifies the RCS client version. client_version_value = Platform "-" VersionMajor "." VersionMinor Platform = Alphanumeric (9 max) VersionMajor = Number (2 char max) VersionMinor = Number (2 char max) Example: client_version=RCSAndrd-1.0	Y	String (15 max), Case-Sensitive
device_type	This indicates the type of device where the client is running.	Y	Possible values: - Tablet - PC - Other
OTP	This is the password provided to the user in step 1. Set to an empty string in case a non-empty token is provided	Y	String (8 Max), Case-Sensitive
friendly_device_name	If provided by the user, a user friendly identification for the device may be passed along that can be used by the network when presenting the user with an overview of their devices.	N, only to be provided if provided by the user	String (30 max before escaping), Case-Sensitive

Table 19: RCS configuration for non-cellular devices: Initial HTTPS request GET parameters

Please note that the initial HTTP request is not required in this case since the header enrichment requirement is not applicable. Therefore, the device implementation/client will directly perform the HTTPS request as presented in Figure X.

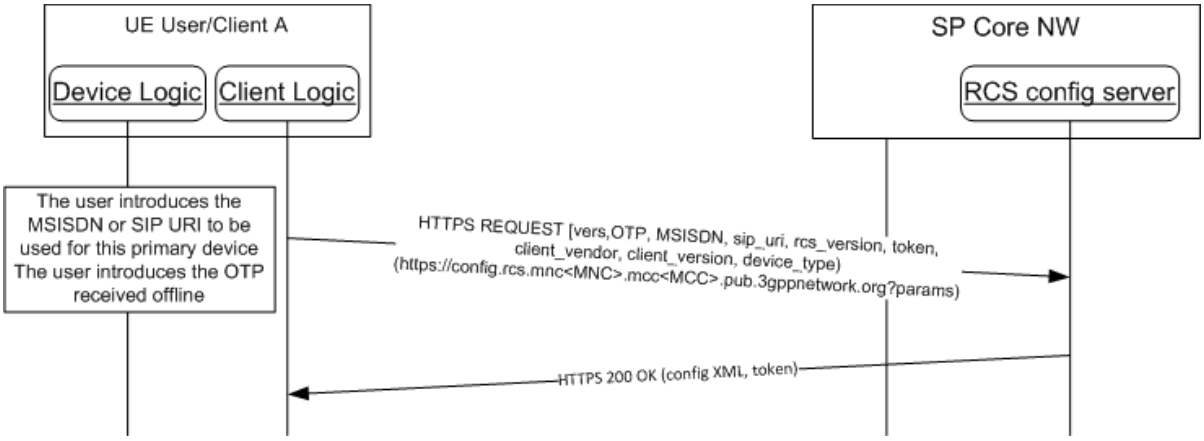


Figure 18: RCS configuration for non-cellular devices with a dedicated identity: initial request

4. As this is a first time request, the token value is empty; the request is then identified as a first time configuration. In this case, and provided the network allows for configuring devices using this mechanism, the HTTP server responds with a HTTP 200 OK response carrying a new cookie (Set-Cookie header) to be used in the subsequent HTTP requests

From this point onwards the procedure is identical to the one described in sections 2.3.3.2.2 and 2.3.3.2.3, however, with token is added as a parameter. If the request is successful, one of the possible 200 OK responses described in section 2.3.3.2.2 is provided. If receiving a full XML and provided the network uses the sip.instance approach for multidevice handling as described in section 2.11, the response shall include the uuid_Value parameters (see Annex A sections A.1.13 and A.2.10 for further reference).

Please note the token shall be stored with the identity so it is not necessary to repeat this procedure for future requests. These values shall be removed together with the rest of the RCS configuration when the device or RCS client are reset.

2.3.3.5.1 Subsequent configuration attempts and life cycle

If the client has access to the token and the identity used for the first time configuration, has a value, the initial request is performed

1. An initial request like in the case of the first time configuration of the primary device is made, this time including the token parameter set to the value received on the previous successful configuration attempt
2. If successful, from this point onwards the procedure is identical to the one described in sections 2.3.3.2.2 and 2.3.3.2.3, however, with the token added as a parameter. If the request is successful, one of the possible 200 OK responses described in section 2.3.3.2.2 is provided. If receiving a full XML and provided the network uses the sip.instance approach for multidevice handling as described in section 2.11, the response shall include the uuid_Value parameters (see Annex A sections A.1.13 and A.2.10 for further reference).

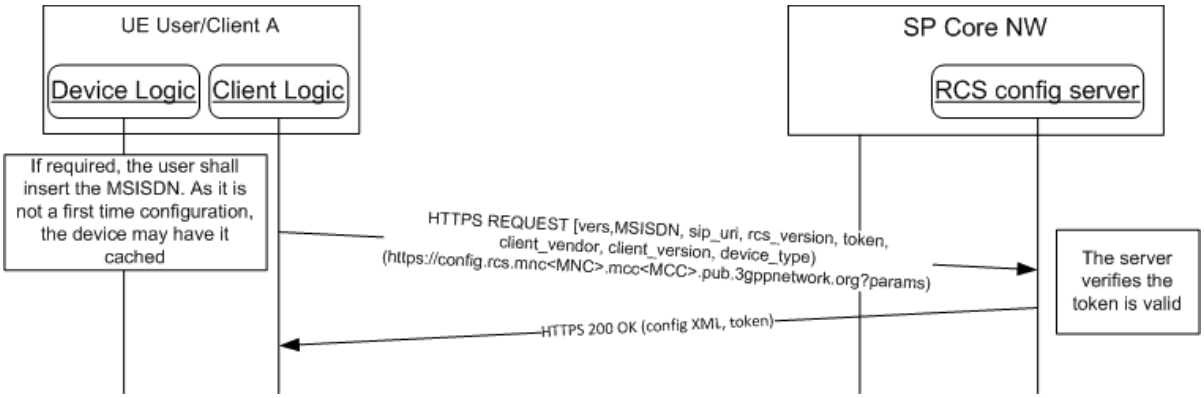


Figure 19: RCS configuration for non cellular devices: Subsequent attempts

If the token and/or the MSISDN are not available (for example the device is reset), then the client shall start a first time configuration as described in section 2.3.3.4.1.

Please note the received token shall be stored with the MSISDN so it is not necessary to repeat this procedure for future requests. These values shall be removed together with the rest of the RCS configuration when the device or RCS client are reset.

2.3.3.5.2 Error handling

In the process of performing a subsequent configuration for additional devices, there are three possible error conditions that the client has to be aware and handle:

1. The MSISDN used is not valid or it is not authorized to get the configuration/make use of the RCS services. In this case, the initial request will be answered with an HTTP 403 FORBIDDEN error and the client shall inform the user of the issue and may offer to retry with a different MSISDN.
2. The token is no longer valid. In this case, the HTTP configuration server replies again with a HTTP 511 NETWORK AUTHENTICATION REQUIRED error. From this moment, the process is equivalent to the first time configuration process after the same error is received.
3. The HTTP server suffers an internal error (HTTP 5XX response coming from the server). This case, the user shall be informed of the circumstance and offered to retry. If retrying, the client shall start the subsequent configuration attempt procedure from the beginning.

2.3.4 Configuration storage on the RCS client

The RCS and, by extension, the IMS configuration should be stored securely on the device and should not be accessible to the user.

As indicated in section 2.4, it should be noted that a precondition to provide access to the RCS functionality is that all the mandatory parameters described in section A.1 must be configured correctly. If any of the required parameters are not configured or configured with an unexpected value, the RCS functionality should be disabled and not be presented or accessible to the user (that is the device behaves as a non-RCS enabled device). In this state, the RCS functionality can only be restored by completing the first-time registration procedure (see section 2.3.1.1; the first-time registration includes the RCS client configuration using one of the procedures described in section 2.3.3).

If an RCS configured device is reset, the RCS client should securely back up the configuration in the device together with the associated IMSI prior to the reset. Please note that this also applies in the event of swapping SIM cards. The configuration associated with the old SIM should then be securely backed up before triggering a first time registration.

The motivation behind the RCS configuration backup is to facilitate the scenario where following a reset or after a SIM swap, the original SIM card is re-introduced into the device. In that instance instead of triggering a first-time registration, the RCS configuration is restored.

In those terminals where as a consequence of the processes mentioned in the previous paragraphs (reset, SIM card swap) the terminal also deletes the contacts (for example a particular Service Provider is enforcing a policy where a SIM swap causes the deletion of the contacts), the associated RCS information (that is the cached capabilities per contact and the RCS contact list) should also be removed. In this case, the RCS information associated with the contacts is not backed up.

The number of configuration backups stored is left to the device's implementation, but shall be at least 2 (for the currently inserted and a previous SIM).

Clients functioning as secondary client sharing the SIM identity used in a user's main device (see section 2.3.3.4) may offer multiple users the possibility to access the services (e.g. by requiring selecting which user to serve when started). In that case the client shall store a user's configuration, Chat histories and call logs in local storage when switching to another account. All private data shall not be accessible to other users. No new configuration requests shall be performed for the stored accounts even when the validity of their configuration expires. A new configuration request shall only be done when a stored configuration is restored because a user has selected to use the account again. If no valid token is available anymore, this may result in a complete first time configuration procedure. When the first time configuration request is successful the stored messages and

communication logs for that user will be made available again as well as the lists of the RCS capable contacts.

2.3.5 Network initiated configuration request

There are use cases (e.g. customer support) where forcing an RCS reconfiguration or a first configuration is required. With OMA-DM, it is possible to push the new configuration data; the HTTP(S) configuration mechanism does not provide such possibility.

The present section presents the enhancements that need to be implemented both on the network side and on the client in order to support a network requested reconfiguration

2.3.5.1 First time configuration initiated via SMS

In this option, the mechanism that will trigger the configuration will be a network originated SMS. Please note that this option is only available to platforms and clients that support the application port addressing (UDH header handling as per [3GPP TS 23.040]).

Regarding the SMS format, the following configuration shall be used:

- DataCodingScheme = 08 (UCS2)
- UserDataHeader = 06 05 04 4074 0000
  a) UDHL length fields=06 05 04,
  b) Destination port: 0x9199 (37273 in decimal)
  c) Source Port: 0x0000 (0 in decimal)
- The SMS content shall be the IMSI associated to the SIM plus the word "rcscfg" preceded by the dash symbol i.e. '-'. For example: If the IMSI is *214011001388741*,
  The value in the text shall be
  214011001388741-rcscfg

When the RCS client receives such request and the IMSI matches the one on the SIM the following actions shall take place:

1. De-register from the IMS network if registered.
2. Start the HTTP configuration described in section 2.3.3.2

2.3.5.2 Reconfiguration initiated via SMS

In this first option, the mechanism that will trigger the reconfiguration will be a network originated SMS. Please note that this option is only available to platforms and clients that support the application port addressing (UDH header handling as per [3GPP TS 23.040]).

Regarding the SMS format, the following configuration shall be used:

- DataCodingScheme = 08 (UCS2)
- UserDataHeader = 06 05 04 4074 0000
  a) UDHL length fields=06 05 04,
  b) Destination port: 0x9199 (37273 in decimal)
  c) Source Port: 0x0000 (0 in decimal)
- The SMS content shall be the Private User Identity (Private_User_Identity parameter in the XML configuration) plus the word "rcscfg" preceded by the dash symbol i.e. '-'. For example: If the private identity is
  214011001388741@ims.mnc001.mcc214.3gppnetwork.org,
  The value in the text shall be
  214011001388741@ims.mnc001.mcc214.3gppnetwork.org-rcscfg

When the client receives such request and the IMPI matches the one in the existing configuration, it shall take the following actions:

1. De-register from the IMS network if registered.
2. Perform a HTTP configuration (as per section 2.3.3.2) setting the version and validity parameters to 0 (i.e. like in the case of a first-time configuration), so it is guaranteed a complete configuration XML is provided by the HTTP configuration server.
3. After the configuration process is completed, the RCS client shall register back to the IMS network with the received settings.

2.3.5.3 Reconfiguration via EUCR request

A reconfiguration can be triggered by the network by sending a EUCR system request with type *urn:gsma:rcs:http-configuration:reconfigure* as specified in section 2.10.

When the client receives such request, it shall take the following actions:

1. Reply the request with a 200 OK response
2. De-register from the IMS network
3. Perform a HTTP configuration setting the version and validity parameters to 0 (i.e. like in the case of a first-time configuration).
4. Register back to the IMS network with the received settings.

2.3.5.4 Interaction with the user during the network initiated reconfiguration

When performing a network initiated reconfiguration and if the user is making use of the service, he shall be notified that the process is taking place and that consequently the service will not be available until the reconfiguration is completed.

2.4 IMS registration

2.4.1 General

Prior to the registration, the device must be configured as described in section 2.3.

The device and IMS core network must follow the SIP registration procedures defined in [3GPP TS 24.229], complemented with the modifications described in this document (e.g. non registration of some feature tags).

NOTE: In particular, the device shall support the nonce storing procedures as defined in [3GPP TS 24.229-rel12] section 5.1 to allow some traffic reduction.

If the device is capable of working in RCS-VoLTE or RCS-VoHSPA mode and is configured to support VoLTE/VoHSPA when available then it must follow the procedures for registration specified in [PRD-IR.92]/[PRD-IR.58] with the changes and additions defined in this document (for example support for GRUU is not required in [PRD-IR.92]/[PRD-IR.58], but it is required for RCS devices). If the device uses the IMS based Conversational Video Service then it must in addition follow the procedures for registration specified in [PRD-IR.94]. In addition, the device is expected to register in IMS with the telephony services provided by the device using the telephony feature tag described in section 2.4.3.

For a device that can be in RCS-VoLTE or RCS-VoHSPA mode i.e. providing and using both RCS and VoLTE/VoHSPA, a common IMS registration is shared by both RCS and VoLTE/VoHSPA.

If the device is in RCS-AA mode or is not configured for VoLTE/VoHSPA as defined in section 2.9.1), it must fulfil the requirements as specified in [PRD-IR.92]/[PRD-IR.58], with the clarification that a device in RCS-AA mode should always register in IMS taking into account Table 3 which indicates which sections apply and provides any necessary clarifications.

When the domain selection has selected IMS voice, the device is in RCS-VoLTE or RCS-VoHSPA mode, and shall not register in non-cellular networks (i.e. it shall not register on Wi-Fi).

NOTE: Parallel IMS registration in VoLTE/VoHSPA and Wi-Fi is out of scope of this specification.

When the domain selection has selected CS voice, the device is in RCS-CS mode, and may remain on the cellular network.

When the device is in RCS-CS mode, it may de-register from IMS on the cellular network and register again through non-cellular access when that is available. This switch to non-cellular access will interrupt any ongoing RCS sessions.

As soon as the domain selection in a device in RCS-VoLTE or RCS-VoHSPA mode is again using IMS voice, it shall attempt to de-register from IMS through the non-cellular access and shall register again using IMS over the cellular network.

For a device that is not configured to use RCS-VoLTE or RCS- VoHSPA mode (see section 2.2.1), the client sends a SIP REGISTER message to the network using the configuration parameters (SIP proxy and other IMS parameters as presented in section A.1.6.2). If supported, the network shall authenticate the message using Single Sign-On (SSO)/General Packet Radio Service (GPRS) IMS Bundled Authentication (GIBA).

The device must use the authentication mechanisms as described in section 2.13.

If the registration is not successful, the user should not be able to access any RCS services and all RCS contacts services/capabilities shall be reported to the user as not available independently of any setting (the IM CAP ALWAYS ON setting presented in Table 85 is ignored for example). When it is the device's network status that prevents the client from registering (e.g. no PS or Wi-Fi connectivity because the device is in "airplane mode") and the IM CAP ALWAYS ON setting is enabled, the chat service may be shown as available even if the client is not registered.

Finally note that a precondition to register is that all of the mandatory parameters presented in section A.1 are correctly configured. This includes those optional parameters that, due to their dependency on the configured value of a mandatory parameter, have become mandatory. If RCS is not the only IMS based functionality available on the device (that is other IMS services are incorporated) however, the precondition does not include having all of the mandatory parameters introduced by section A.1.6 correctly configured as in that scenario this configuration may have been obtained through other means.

2.4.2 Procedures for multidevice handling: GRUU and sip.instance

The device shall support using Globally Routable User agent URIs (GRUUs) to uniquely address each RCS client residing on different devices as specified in [3GPP TS 24.229] taking into account the clarifications given below.

When the user agent generates a REGISTER request (initial or refresh), it shall include the Supported header field in the request. The value of that header field shall include "*gruu*" as one of the option tags. This indicates to the registrar for the domain that the User Agent (UA) supports the GRUU mechanism.

In each contact included in the REGISTER request, the client shall include a "*sip.instance*" tag, whose value is the instance ID that identifies the user agent instance being registered. As network support for GRUU is not mandatory, sip.instance can be used instead. An RCS client will use a public GRUU if provided by the network, but there is no requirement in RCS for a device to use a temporary GRUU.

If the RCS client type is embedded (as described in section 2.2.2) and has access to the device IMEI, then sip.instance shall be the IMEI value as per [3GPP TS 24.229]. Otherwise, the value of sip.instance shall use either:

- The value provided as part of the device/client configuration (uuid_Value, as described in Annex A sections A.1.13 and A.2.10) shall be used. In this case, the network shall follow one of the algorithms described in [RFC4122], or,
- If the uuid_Value is not provided as part of the configuration (parameter not present in the configuration or present but with an empty value), the UUID (Universal Unique Identifier) shall be generated as per [RFC4122] section 4.2 and in all cases, must not be modified over time.

If the REGISTER response is a 2xx and the network supports GRUU, each Contact header field will contain a "pub-gruu" conveying the public GRUU for the user agent instance. The GRUU support is not mandatory for the Service Providers. Therefore user agents shall not always expect to receive a GRUU from the registrar.

2.4.2.1 Additional clarifications on sip.instance usage for multidevice support

When an RCS client is configured to use sip.instance, all SIP requests and responses that contain a Contact header will carry the sip.instance.

When an RCS client is required to ensure that a generated SIP request is sent back to the same device that was identified through sip.instance, a new *Accept-Contact* header is added carrying only the sip.instance tag and instance identifier value as well as the tags explicit and require described in [RFC3841].

Regarding the support of routing based on the value of a sip.instance feature tag by an IMS core, there are two possible scenarios:

1. If the IMS core supports the procedures described in [RFC3841], then any SIP request with an *Accept-Contact* header that addresses a specific RCS device is only received by that specific instance/device.
2. If not, then all the RCS clients registered using the same IMS identity will receive the SIP request. Consequently, an RCS client supporting the *sip.instance* procedures shall respond to the invite with a 486 BUSY HERE if the identifier value of the *sip.instance* tag included in the *Accept-Contact* header of that incoming SIP request does not match theirs.

2.4.3 Telephony feature tag

RCS defines a telephony feature tag used to indicate to the network which set of telephony methods are supported by the device. The feature tag is set in the Contact header at registration with possible sets of values to include: “*none*”, “*cs*”, “*cs,volte*”, “*cs,volte,vohspa*”, “*cs,vohspa*”, "*vohspa*", "*volte*" or "*vohspa,volte*".

The feature tag is defined as +g.gsma.rcs.telephony=<set of values>.

For example,

- for a device that does not support any telephony services, the feature tag would be set as follows: +g.gsma.rcs.telephony=”none”.
- for a device that supports only CS as telephony method, the feature tag would be set as follows: +g.gsma.rcs.telephony=”cs”.
- for a device supporting both CS and VoLTE, it would be set as follows: +g.gsma.rcs.telephony=”cs,volte”.

2.4.4 Services feature tags

2.4.4.1 Service related feature tags at IMS registration as per service specifications endorsed by RCS

The client shall include the feature tags related to the authorized/enabled services in the REGISTER request as per the relevant service specifications (e.g. *+g.oma.sip-im* as per [RCS5-SIMPLEIM-ENDORS]).

2.4.4.2 File Transfer via HTTP feature tags at IMS registration

When File Transfer via HTTP is enabled (the configuration parameters *FT HTTP CS URI, FT HTTP CS USER* and *FT HTTP CS PWD* defined in section A.1.4 are correctly set), the client shall include the File Transfer via HTTP IARI defined in Table 23 in the REGISTER request.

2.4.4.3 Geolocation PUSH feature tags at IMS registration

When Geolocation PUSH is enabled (see PROVIDE GEOLOCATION PUSH in section A.1.7.2), the client shall include the Geolocation PUSH IARI defined in Table 28 in the REGISTER request.

2.4.4.4 RCS IP Call feature tags

When RCS IP Voice or Video Calls are enabled for any cellular access (see PROVIDE RCS IP VOICE CALL and PROVIDE RCS IP VIDEO CALL in section A.1.14), an RCS client shall when registering in the IMS over a cellular access include the feature tags for RCS IP Calls according to Table 3.

When RCS IP Voice or Video Calls are enabled for Wi-Fi (see PROVIDE RCS IP VOICE CALL and PROVIDE RCS IP VIDEO CALL in section A.1.14), an RCS client shall when registering in the IMS over Wi-Fi access include the feature tags for RCS IP Calls according to Table 3. There is no need for the device to remove the RCS IP Voice/Video service capability once it has registered with it.

RCS defines the RCS IP call feature tags used to indicate the wanted behaviour as listed in Table 20.

RCS service	Tags
RCS IP Voice Call (strong preference for no breakout (IP end to end))	+g.gsma.rcs.ipcall; +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel"
RCS IP Video Call (strong preference for no breakout (IP end to end))	+g.gsma.rcs.ipcall; +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel";video
RCS IP Video Call where video media cannot be removed by the user (strong preference for no breakout)	+gsma.rcs.ipcall;+g.gsma.rcs.ipvideocallonly;+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel";video

Table 20: RCS IP Voice/Video Call feature tags and resulting

See Table 3 for the media feature tags required at IMS registration and at session setup.

NOTE: IP Voice/Video Call identification for calls without the RCS IP Call tags is as per [PRD-IR.92]/[PRD-IR.94]. In other words it is up to a service provider to decide upon call breakout.

2.4.4.5 Extension feature tags

An Extension is enabled or disabled via operator specific means. For example, an operator may make use of the EUCR mechanism as describe in section 2.10.4 to enable or disable an Extension. Only enabled Extensions are allowed to use the RCS infrastructure.

If an RCS Client supports Extensions that are allowed to use the RCS infrastructure, then when registering in the IMS the RCS Client shall include

- The RCS Extension to Extension data channel ICSI(s) (as per section 3.12.4.2.2.1) if at least one enabled Extension uses this service, if and only if the configuration parameter ALLOW RCS EXTENSIONS (as defined in section A.1.16) is set to 1.
- The IARIs of the enabled Extensions as defined in Table 30, for the Extensions using the RCS Extension to Extension service or any RCS service, if and only if the configuration parameter ALLOW RCS EXTENSIONS (as defined in section A.1.16) is set to 1.

When an Extension is enabled after initial registration, a re-REGISTER request with the new IARI included in the list of media feature tags carried in the Contact header shall be sent out immediately.

When an Extension is disabled by the operator, the Extension's IARI is removed from the list of media feature tags carried in the Contact header of the SIP REGISTER request.

NOTE: To remove an IARI, the device may wait till the next scheduled refresh re-REGISTER request or may issue a re-REGISTER request immediately.

2.4.5 Registration flows

Prior to the first IMS registration, it is necessary to provision the user on the network (e.g. by auto-provisioning) and to configure the device with the required settings (see chapter 2.3). When the provisioning and device configuration phase is completed, the IMS registration takes place. Figure 20 shows the registration flow which applies for all authentication mechanisms covered in Section 2.13. After successful IMS registration, the device performs a capability and new user discovery procedure as described in Section 2.6.

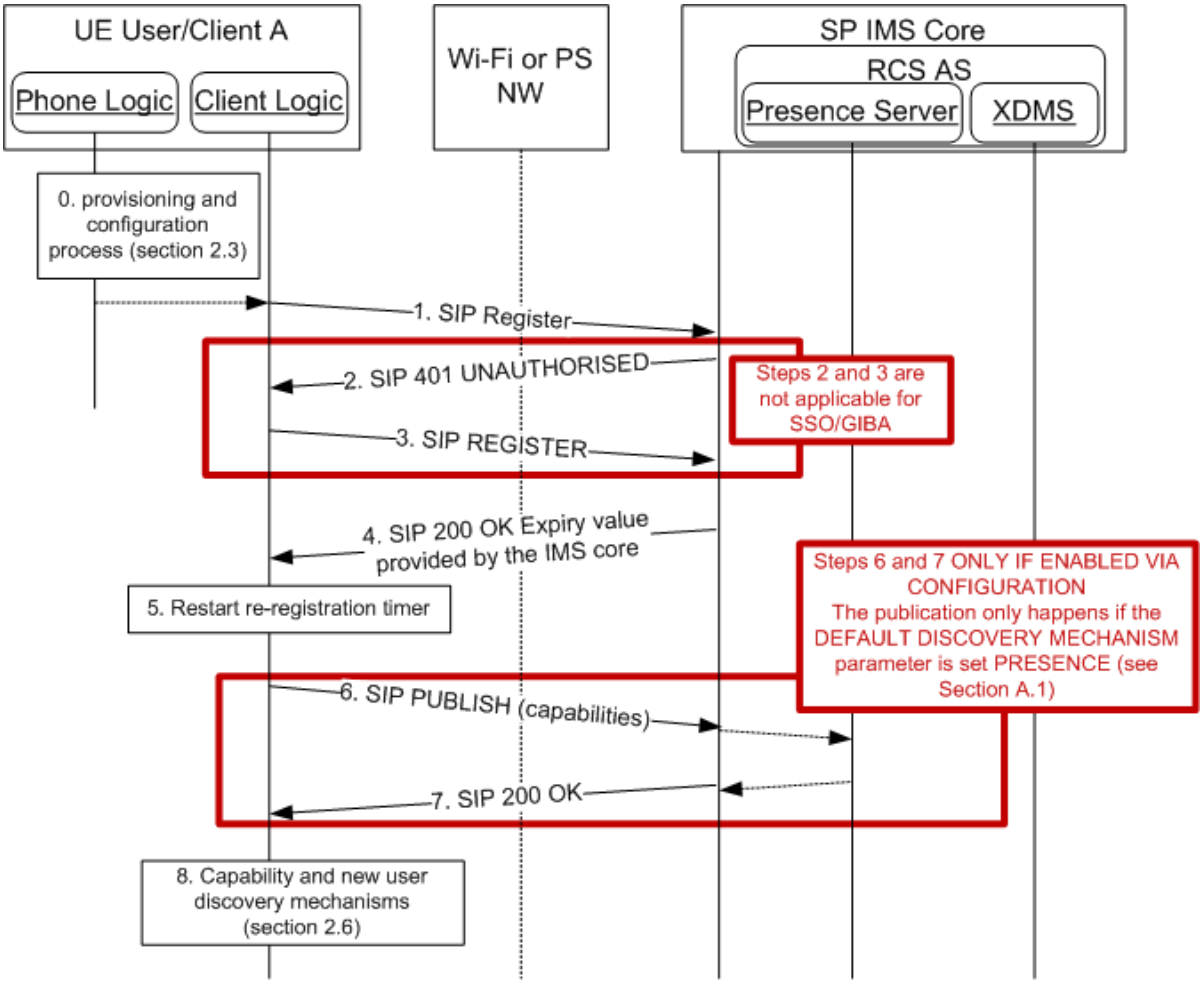


Figure 20: IMS registration flow

Once registered the re-registration timer is initiated which is used to refresh the registration before it expires. If a presence based capability check is used (based on the DEFAULT DISCOVERY MECHANISM parameter specified in Table 91 in section A.1.10 see section 2.6), the device shall also publish its capabilities once it has registered.

NOTE: As stated in section 2.13, if SSO/GIBA authentication fails, Digest authentication shall be tried.

2.4.6 P-CSCF discovery

Prior to any initial IMS registration the client shall discover the IP address of the P-CSCF as defined in [3GPP TS 23.228].

The P-CSCF discovery procedure shall be applied in accordance with the mode of operation of the RCS device:

- devices in RCS-VoLTE mode P-CSCF shall select the P-CSCF as defined in [PRD-IR.92]
- devices in RCS-VoHSPA mode P-CSCF shall select the P-CSCF as defined in [PRD-IR.58]
- Devices in RCS-CS mode and RCS-AA mode shall select the P-CSCF from the LBO_P-CSCF_Address node of the IMS management object.

If the P-CSCF AddressType of the P-CSCF address in the IMS management object indicates "FQDN" the device shall resolve the Fully Qualified Domain Name (FQDN) as defined in [RFC3263].

For the protocol selection in [RFC3263], the device shall

- take the SIP transport protocol settings in the "other" node of the device management object (as defined in section A.2.10) into account if SIP Digest is to be used for authentication, i.e. SIPoUDP, SIPoTLS or SIPoTCP depending on the access network type (PS or Wi-Fi).
- ignore the SIP transport protocol settings in the "other" node of the device management object (as defined in section A.2.10) if AKA is to be used for Authentication. The device selects either UDP or TCP as defined in [3GPP TS 24.229].

If the P-CSCF discovery results in a list of P-CSCF addresses then the device shall select a new P-CSCF address for any initial registration in accordance with the priority indications (e.g. weight and priorities in DNS SRV) to support load distribution in the network.

Although the P-CSCF discovery method differs for the operation modes of a device no actions are required by the device for an active registration when changing the mode, e.g. from RCS-VoLTE to RCS-CS mode while leaving LTE coverage.

The network operator should ensure that the P-CSCF addresses advertised by the different discovery methods are consistent.

2.4.7 IMS Flow Set Management

IMS flow set is defined in [3GPP TS 24.229]. It refers to the "flow" defined by the combination of transport protocol, client IP address and port and P-CSCF IP address and port used by the client and the network to exchange all SIP signalling related to a single IMS registration. This section details the requirements for an RCS client to manage the IMS flow set (i.e. a single Registration) in the network.

2.4.7.1 REGISTER Request Handling

The RCS client shall make use of the registration procedures as defined in [3GPP TS 24.229].

Note: [3GPP TS 24.229] specifies that the REGISTER shall be sent to the IP address and port obtained via the discovery procedure. If the device was unable to obtain a specific port, then the default port as specified in [RFC3261] will be used.
The client will send subsequent REGISTER and non-REGISTER requests to the IP address that is used for the initial REGISTER, unless the security mechanism requires the use of negotiated ports for the exchange of protected messages.

2.4.7.2 Non REGISTER Request Handling

The RCS client shall make use of the procedures for methods excluding the REGISTER method as defined in [3GPP TS 24.229].

The following addition applies to [3GPP TS 24.229] section 5.1.2A.1.1:

- The proper preload route header for methods excluding the REGISTER method shall be built only with the IP address learnt through the P-CSCF discovery procedure, i.e. a FQDN must not be used.

2.4.7.3 IMS Flow Set Termination

The RCS client should ensure that an IMS flow set is released in the network before the conditions for the existence cease to exist, e.g. prior to the release of the bearer the IMS flow set makes use of.

The IMS flow set shall be terminated by the client by sending a de-registration request to the network using the IMS flow set to be terminated. If there is one or more ongoing session on the IMS flow set, these shall be released first.

2.4.7.4 Loss of Connection to P-CSCF

If the connection to the P-CSCF fails (e.g. TCP time-out) the RCS client should select another P-CSCF address from the list of addresses obtained during the P-CSCF discovery in accordance with their priority indication.

If the P-CSCF discovery is based on the IMS management object and it contains one or more FQDNs, then the client shall invoke the [RFC3263] FQDN resolution anew. A different P-CSCF address shall be selected from the name resolution result in accordance with their priorities and weights.

The client shall then send a new initial registration using the new discovered P-CSCF address.

2.4.7.5 Loss of Connectivity

If a RCS client discovers that connectivity has been lost then it should attempt to re-establish the connection.

When connectivity has been resumed then;

- If the IP address has been changed and the transport protocol setting for the new connection (as derived from the Management Object defined in section A.2.10 for the new access network type) is the same as for the lost connection and the IMS registration is not yet expired, a client in RCS-CS or RCS-AA mode shall perform a new initial registration to the P-CSCF address of the last IMS flow set in use.
- If the IP address has not been changed and the IMS registration is not yet expired, the client shall perform a re-registration using the existing IMS flow set if the IP address has not been changed and the IMS registration is not yet expired. To minimize the network impact in cases of unstable connectivity conditions the client should hold a minimum re-registration time in which no such re-registration requests are sent. The minimum re-registration time should be typically in the range of 3-5 minutes.
- In all other cases the client shall perform a new P-CSCF discovery and a new initial registration.

  NOTE: The registration or re-registration may trigger delivery of messages stored in the network during the absence of connectivity.

2.4.7.6 Detection of Connection Loss in RCS Clients with no Bearer Control Capabilities

RCS client implementations may have no capability to identify the cause of a connection loss due to missing bearer control capabilities.

These clients should identify the cause of a loss of connectivity via the following procedure.

- If the client detects a connection loss during a P-CSCF signalling interaction (e.g. TCP time-out), then it shall attempt the procedure defined in section 2.4.7.4.

- Only if a new IMS flow set is established with an alternative P-CSCF the client shall release the IMS flow set used for the old P-CSCF locally.
- If the connection establishment to the alternative P-CSCF or other targets in the network fails (e.g. DNS Server) then the client shall assume loss of connectivity and act as defined in section 2.4.7.5.
- If the client detects a connection loss during network interactions other than signalling with the P-CSCF (e.g. media connection, auto-configuration server) then the client shall assume loss of connectivity.

2.4.8 Loss of Registration

When the client receives a SIP response to a non-REGISTER request that is either:

- 403 Forbidden without a warning header, or
- 504 Server Timeout containing a P-Asserted-Identity URI matching a URI received during registration in Service-Route or Path header field and containing a 3GPP IM CN subsystem XML body with the <alternative-service> child element with the <type> child element set to “restoration” and the <action> child element set to “initial-registration”

(indicating loss of registration due to change of IP, expiration, network problem), the client shall attempt to register again using the procedure in section 2.4.5. When successful the client shall resend the request that caused the error response. If this fails for 5 consecutive retries though, no further attempt shall be made and an error should be shown to the user. For all services except One-to-One Chat, the retry procedures will also be stopped if it takes longer than 5 seconds. Also in that case an error message should be shown to the user.

NOTE: On receiving a 403 Forbidden response a client may before re-Registration first attempt to send a SIP request to his own URI and only re-Register if that request results in a 403 Forbidden response.

2.5 Addressing and identities

2.5.1 Overview

Telephone numbers in the legacy address book must be usable (regardless of whether RCS contacts have been enriched or not) for the identification of contacts of incoming and outgoing SIP requests.

Also, RCS users, especially in Enterprise segments, may be assigned a non MSISDN based identity. The RCS client would in that case be provisioned with only the appropriate SIP URI parameter as seen in section A.1.6.3, leaving the tel URI parameter empty.

Consequently, an RCS enabled terminal’s address book should also be able to store alphanumeric SIP URIs as part of a contact’s details.

NOTE1: the handling of identities described in this section applies also to IP Voice Calls [PRD-IR.92] and [PRD-IR.58]. The functionality described here comes in addition to the functionality described in the related Permanent Reference Documents (PRDs), but not in conflict with them, e.g. the alias handling described in section 2.5.3.3.

NOTE2: the identification in Common Profile for Instant Messaging (CPIM) headers is discussed in section 3.3 and 3.4.

2.5.2 Device Incoming SIP Request

2.5.2.1 From/P-Asserted-Identity

For device incoming SIP requests, the address(es) of the contact are, depending on the type of request, provided as a URI in the body of a request or contained in the *P-Asserted-Identity* and/or the *From* headers. If the *P-Asserted-Identity* header is present, the *From* header will be ignored. The only exception to this rule is when a request for Chat or Standalone Messaging includes a *Referred-By* header (it is initiated by Messaging Server for example in a store and forward use case as described in 3.3.4.1.4), thereby the *Referred-By* header should be used to retrieve the originating user instead.

The receiving client will try to extract the contact's phone number out of the following types of URIs:

- tel URIs (telephone URIs, for example *tel:+1234578901*, or *tel:2345678901;phone-context=<phonecontextvalue>*)
- SIP URIs with a "user=phone" parameter, the contact's phone number will be provided in the user part (for example *sip:+1234578901@operator.com;user=phone* or *sip:1234578901;phone-context=<phonecontextvalue>@operator.com;user=phone*)

Once the MSISDN is extracted it will be matched against the phone number of the contacts stored in the address book. If the received URI is a SIP URI but does not contain the "*user=phone*" parameter, the incoming identity should be checked against the SIP and tel URI address of the contacts in the address book instead.

If more than one *P-Asserted-Identity* is received in the message, all identities shall be processed until a matched contact is found.

2.5.3 Device Outgoing SIP Request

2.5.3.1 Identification of the target contact

If the target contact contains a SIP or tel URI the value shall be used by the RCS client when generating the outgoing request even if an MSISDN is also present for the contact. This applies to the SIP Request-URI and the "*To*" header (as defined in [3GPP TS 24.229]) for 1-to-1 communication, including the URIs used in the recipient list included in outgoing SIP requests for Group Chat.

If no SIP or tel URI is present the RCS client shall use the telephone number (in local format for example *0234578901* or international format *+1234578901*) set in the address book or a dial string entered by the user.

If the target number is an international-format telephone number, the device shall be able to send it as tel URI (for example "*tel:+12345678901*") as defined in [RFC3966] or as SIP URI (for example *sip:+12345678901@domain;user=phone*) with the user parameter set to "phone" as defined in [RFC3261]. The format used is configurable on the device (see tel or SIP URI – international in Table 88 in section A.1.6.3) and shall be set according to the Service Provider's requirements or constraints related to national regulatory framework of SIP-SIP interconnection (the Service Provider will provide this choice during customization). If none of the above constraints apply, the use of tel URI is recommended as the domain name of the SIP URI is not significant.

If the target number is a non-international format telephone number, the RCS client shall be able to send it as tel URI or as SIP URI (the user parameter should be set to "phone") with a phone-context value set as defined in [3GPP TS 24.229] for home local numbers (for example "*tel:0234578901;phone-context=<home-domain-name>*"). Similar to the international number case, whether a tel URI or a SIP URI is used, is configurable on the device (see tel or SIP URI - for non- international format in Table 88 in section A.1.6.3) and shall be set according to the Service Provider's requirements or constraints related to

national regulatory framework of SIP-SIP interconnection. If none of the above constraints apply, the use of tel URI is recommended.

2.5.3.2 Self-Identification to the network and the addressed contact

When generating an outgoing non-REGISTER request, the RCS client shall populate the *From* header field and may populate the *P-Preferred-Identity* header field with a SIP or tel URI which has been received in the *P-Associated-URI* header field returned in the 200 OK to the SIP REGISTER. If both a SIP URI and a tel URI are available to the RCS client, the tel URI should be used.

2.5.3.3 User alias

The user shall be able to specify an alias or a username for RCS services. This information will be sent when establishing a communication service with another user so they are able to receive additional information (i.e. beyond than just a MSISDN), if the originating user is not in the receiver's address book. This scenario will likely be common with Group Chat sessions.

This alias information will be set in the *From* header of the SIP request as the display name and in a Group Chat also in the CPIM *From* header as the formal name.

When receiving a request, the RCS client device shall follow the rules explained in section 2.5.2.1 and extract the MSISDN or SIP URI. To avoid spam and identity manipulation, the receiver shall check the identity of the calling user against the address book. If the user is not in the address book, the alias information must then be used to provide more information about the calling user while clearly displaying in the User Interface (UI) that the identity is unchecked and it could be false. Otherwise the name of the contact in the address book shall be used instead.

2.6 Capability and new user discovery mechanisms

2.6.1 Capability discovery

The capability or service discovery mechanism is vital to RCS. The capability discovery is a process which enables a user to understand the subset of RCS services available to access and/or communicate with their contacts, at certain points in time.
The RCS specification provides two alternative mechanisms to perform the capability discovery:

- SIP OPTIONS exchange (section 2.6.1.1):
  - The SIP OPTIONS end-to-end message is used both to query the capabilities (services which the other user has available) of the target contact and to pass the information about which capabilities are supported by the requester. Using this method, both users get updated information in a single transaction. However, users are allowed to hide their service capabilities to some undesired contacts. To this end, clients may support a locally stored capability blacklist which could be configured manually by the user or be generated automatically by setting any contacts that do not exist in the user's address book as the blacklisted contacts. This blacklist is different with the blacklist that is used for chat, file transfer and content share services. If a SIP OPTIONS request is received from a capability blacklisted contact, the client should answer the request with an empty 200 OK which does not include any of the service tags used by RCS services (see Table 34).
  - This method requires a specific application server (Options-AS) in the network to provide multidevice support and, potentially, include optimizations.

- Presence (section 2.6.1.2):
  - In this case, instead of performing an end-to-end transaction, the capabilities are queried against a server using the standard OMA SIMPLE Presence procedures which are described in detail in section 2.6.1.2.
  - Consistent with the previous paragraph and the OMA SIMPLE Presence procedures, this method requires both a Presence and a XDM server in the network.

The default mechanism is configured in the device using the configuration parameter CAPABILITY DISCOVERY MECHANISM (see Annex A section A.1.10).

In accordance with the principle of interoperability between RCS networks and devices, two mechanisms are provided to secure the interoperability between the mechanisms presented before:

- Coexistence based in a common device stack (section 2.6.1.3):
  - The interoperability is provided via a device implementation and, consequently, no additional network elements are required.
  - The principle of interoperability is that all devices support SIP OPTIONS exchange either as a default or a device fall-back mechanism (when the presence query fails for a particular user)
- Coexistence based in network interworking (section 2.6.1.3.3):
  - Network Interworking is required between Service Providers that do not support SIP OPTIONS exchange (as the default method or as a device fall-back mechanism) and those Service Providers that use SIP OPTIONS as the default discovery mechanism.
  - Interoperability is achieved by deploying a network based interworking function which translates requests and responses between the SIP OPTIONS and presence-based capability discovery mechanisms.

To guarantee that Service Providers choosing the network interworking approach do not experience situations whereby the device fall-back mechanism to SIP OPTIONS occurs, a new parameter (CAPABILITY DISCOVERY VIA COMMON STACK) is defined. The device fall-back mechanism only occurs if this parameter is set to 1 (see Annex A sections A.1.10 and A.2.8 for further reference).

2.6.1.1 Capability discovery process through SIP OPTIONS message

One of the available mechanisms for capability discovery is based on the exchange of a SIP OPTIONS request, a peer-to-peer message exchanged between clients.

When a SIP OPTIONS message is sent from User A to User B, User A shall handle the response as described in the following table:

Response	User B was a known RCS user before	User B was not a known RCS user before
200 OK including at least, one of the tags assigned to the RCS Services (see Table 34) Returned when User B is an RCS user and is currently registered	User B remains an RCS user The capabilities returned in the 200 OK response (using tags as described in Table 34) are considered as the current communication options with User B	User B is marked as an RCS user The capabilities returned in the 200 OK response (using tags as described in section Table 34) are considered as the current communication options with User B

200 OK not including any of the tags used by RCS services (see Table 34) Returned when User B is registered, but not with an RCS client or User A is in the capability blacklist of User B	User B is not considered as an RCS user any longer Only the non-RCS communication services (e.g. voice calls, SMS, MMS, etc.) are indicated as available[4]	No change in User B's status Only the non-RCS communication services (e.g. voice calls, SMS, MMS, etc.) are indicated as available
480 TEMPORARY UNAVAILABLE or 408 REQUEST TIMEOUT Returned by the network if User B is an IMS (and potentially thus an RCS) user, but is currently not registered	User B remains an RCS user but only the capabilities available to an offline contact are offered	No change in User B's status Only the non-RCS communication services (e.g. voice calls, SMS, MMS, etc.) are indicated as available
404 Not Found or 604 Does Not Exist Anywhere or any other Final Response[5] returned by the network when User B is not provisioned as an IMS user	User B is not considered as an RCS user any longer Only the non-RCS communication services (e.g. voice calls, SMS, MMS, etc.) are indicated as available	No change in User B's status Only the non-RCS communication services (e.g. voice calls, SMS, MMS, etc.) are indicated as available
Any other Final response returned by the network	User B remains an RCS user with unchanged capabilities. NOTE: The client treats the final response as described in [3GPP TS 24.229].	No change in User B's status

Table 21: Options response handling

In some cases sending an OPTIONS request is not required as the last SIP OPTIONS exchange took place just before the communication was set up (e.g. to send a SMS message, the user went to the address book, selected a user [SIP OPTIONS exchange takes place] and chooses to send a SMS message).

[4] Note that this means that an AS like the OPTIONS-AS described in section 2.6.1.1.5 would have to include the IM capability in the response if the user has multiple devices sharing the same IMS identity some of which are not RCS capable. When including this tag though in situations where none of the RCS capable devices is online, it shall also include the *automata* tag defined in [RFC3840] to indicate that this response does not originate from an end user device.

[5] Please note that the response provided may depend on the network configuration. For simplicity, the present document assumes in the following sections that the response provided by the Service Provider core network is always 404 NOT FOUND, however, the previous statement should be taken into account.

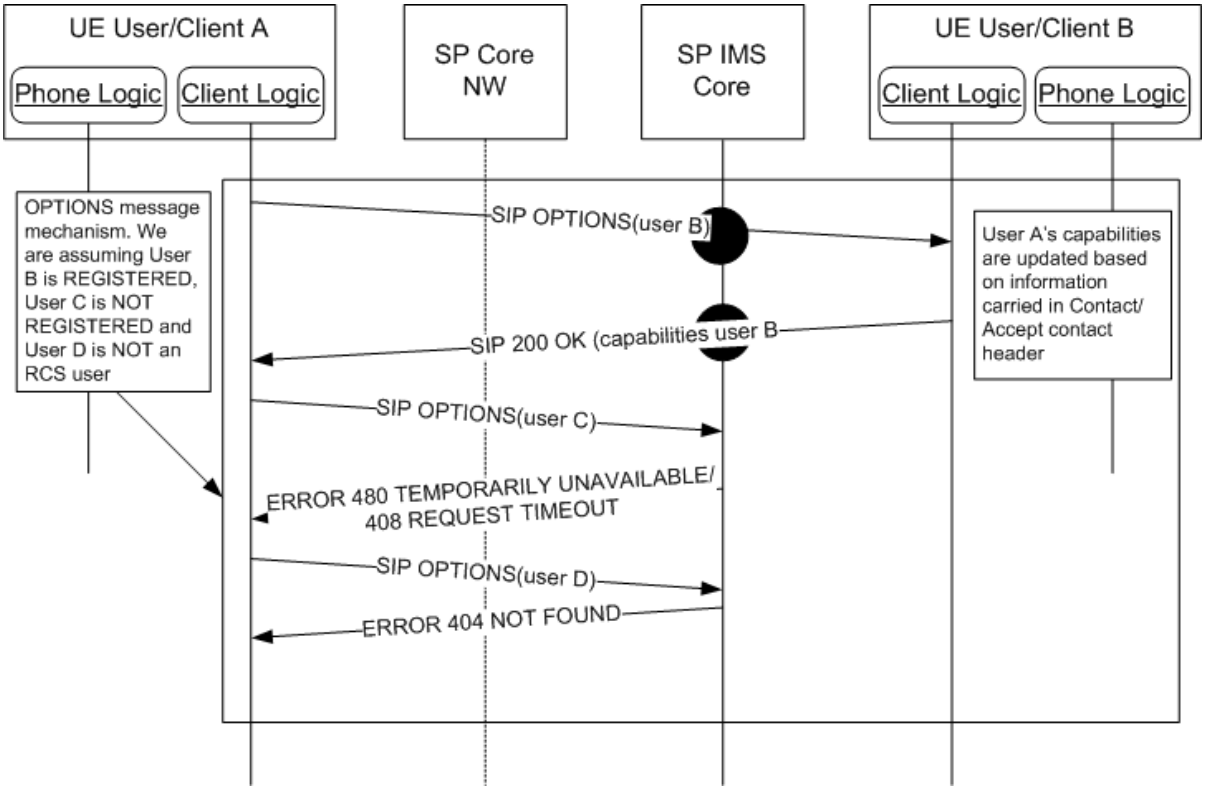


Figure 21: Capabilities discovery via SIP OPTIONS message

2.6.1.1.1 SIP OPTIONS message extension to support capability discovery

The RCS (Release 1 to 4) specifications only provide a mechanism to exchange the capability status (based on a SIP OPTIONS exchange) related to the Image and Video Share services during a call (associated with the capability query procedure described in [PRD-IR.74] and [PRD-IR.79]). This mechanism is based on the use of tags transported in the *Contact* header field for the SIP OPTIONS and its responses:

- The tags corresponding to the set of functionalities supported by the requesting terminal at the time this request is made are carried in the *Contact* header field of the SIP OPTIONS message.
- The tags corresponding to the subset of the functionalities that are supported by the receiver are included in the *Contact* header of the 200 OK responses.

When the SIP OPTIONS is sent as part of an ongoing voice call, as per [PRD-IR.74] and [PRD-IR.79], the Accept-Contact header shall be handled as described in [PRD-IR.74] and [PRD-IR.79].

As described in [PRD-IR.74] and [PRD-IR.79], to have a Session Description Protocol (SDP) body in an OPTIONS request message is optional. It is not encouraged behaviour to insert it into this message. In RCS, the SIP OPTIONS request shall NOT contain an SDP body.

The mechanism described above is extended to be used not only for the exchange of capabilities for real-time services but also to query in real time to exchange the capabilities/services supported by both the requester and the receiver.

2.6.1.1.2 Extensions to the existing tags

Consequently with the RCS Release 1-4 specifications, the following tags can be employed to identify Image and Video Share service capabilities during a call:

RCS service	Tag
Image Share	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.gsma-is"
Video Share	+g.3gpp.cs-voice

Table 22: Standard RCS Release 1-4 SIP OPTIONS tags

When used in SIP OPTIONS exchanges these Image and Video Share capabilities can only be sent during an active call and are included only if the exchange takes place between the users in the active call. However Broadband Access devices should include these capabilities in an OPTIONS response even if they are not in an active call.

To support the full service discovery functionality presented in this document, it is necessary to extend the tag mechanism by adding the following service tags:

- As interoperability between the different technical implementations for Chat and File Transfer services is assumed, the following tags are employed for the Chat and File Transfer services, with a new one added for the store and forward Group Chat service:

RCS service	Tag
Chat	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcse.im"
Full Store and Forward Group Chat	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.fullsfgroupchat"
File Transfer	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcse.ft"
File Transfer Thumbnail	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.ftthumb"
File Transfer Store and Forward	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.ftstandfw"
File Transfer via HTTP	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.fthttp"

Table 23: SIP OPTIONS tags for Chat and File Transfer

- Add a tag for IP based standalone text and multimedia messaging :

RCS service	Tag
IP Based Standalone messaging	+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.oma.cpm.msg,urn%3Aurn-7%3A3gpp-service.ims.icsi.oma.cpm.largemsg"

Table 24: SIP OPTIONS tag for standalone messaging

- Add a tag for the video sharing outside of a call service:

RCS service	Tag
Video Share outside of a voice call	+g.3gpp.iari-ref=”urn:urn-7:3gpp-application.ims.iari.gsma-vs"

Table 25: SIP OPTIONS tag for video sharing outside a call

- Add a tag for Social Presence Information:

RCS service	Tag
Social presence information	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcse.sp"

Table 26: SIP OPTIONS tag for Social Presence Information

- Add tags for IP Voice and Video Call services:

RCS service	Tag
IP Voice Call (as per MMTEL)	+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel"
IP Video Call(as per MMTEL)	+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel";video
RCS IP Voice Call	+g.gsma.rcs.ipcall
RCS IP Video Call	+g.gsma.rcs.ipcall;video
RCS IP Video Call where video media cannot be removed by the user	+g.gsma.rcs.ipvideocallonly

Table 27: SIP OPTIONS tags for IP Voice and Video Call

Note also that when a device supports both IP Voice Call and IP Video Call, the feature tag *+g.3gpp.icsi-ref="urn:urn-7:3gpp-service.ims.icsi.mmtel"* and *+g.gsma.rcs.ipcall* are only included once in the OPTIONS request/response.

A device shall provide in the SIP OPTIONS requests and responses only one of the *RCS IP Voice Call*, *RCS IP Video Call* and *RCS IP Video Call where video media cannot be removed* capabilities depending on whether according to the PROVIDE RCS IP VOICE CALL and PROVIDE RCS IP VIDEO CALL configuration parameters defined in section A.1.14 in the currently used radio technology respectively only RCS IP Voice Calls, both RCS IP Voice Calls and RCS IP Video Calls or only RCS IP Video Calls are allowed. Due the combining of the capabilities of different devices (see section 2.6.1.1.5) it may happen though that multiple of those capabilities are received. If the RCS IP Video Call where video media cannot be removed capability is received in combination with the RCS IP Video Call and/or the RCS IP Voice Call capability, towards that contact both RCS IP Voice Call and RCS IP Video Calls may be successfully initiated as described in sections 3.8 and 3.9 when those services are allowed according to the local configuration and used radio technology.

NOTE1: A UE may support both IP voice/video call and RCS IP voice/video call services. In that case, all tags correspondent to all supported services shall be provided.

NOTE2: RCS IP Calls and IP Calls per MMTEL can technically interoperate. Therefore an RCS IP Call client is allowed to answer an incoming IP Call (and vice versa), and shall process the received capability information accordingly.

If the device is configured to support only the RCS IP Video Call services as described in section A.1.14 (i.e. RCS IP Voice Call initiation is not allowed and a downgrade by the user of an RCS IP Video Call to an RCS IP Voice Call is not allowed), it shall include the

+g.gsma.rcs.ipvideocallonly tag described in section 2.4.4 in the OPTIONS exchanges when in coverage where RCS IP Video Calls are supported.

The applicable RCS IP Call feature tags are only included in the OPTIONS exchanges when the device is in coverage where it is configured to support RCS IP Voice and Video Calls.

NOTE3: In case of interworking between two Service Providers, the validity of the IP Video Call capability tag highly depends on the end-to-end interconnection chain.

- Add tags for the Geolocation services:

RCS service	Tag
Geolocation PUSH	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.geopush"
Geolocation PULL	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.geopull"
Geolocation PULL using File Transfer	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.geopullft"

Table 28: SIP OPTIONS tags for geolocation services

Unless specified in other sections (e.g. section 2.4.4), the new tags defined in this section are defined for use in SIP OPTIONS exchanges only and the standard tags defined in the supporting PRDs and endorsement documents shall be used to identify the services in the rest of relevant SIP transactions (e.g. *+g.oma.sip-im* for Chat implementation based on OMA SIMPLE IM as per [RCS5-SIMPLEIM-ENDORS]). It should also be noted that in some cases, the tags employed in the SIP OPTIONS exchange match the standard tags.

Note that, a device should also add to the Contact header field the same feature tags used at SIP Registration if not already included in the OPTIONS request/response for capability exchange and if they are part of the capabilities supported by the device at this time. This mainly applies to the *+g.oma.sip-im* feature tag which is used at SIP Registration and in SIP transactions but not used to identify a service capability.

Finally, it should be taken into account that when several IMS Application Reference Identifier (IARI) tags or several ICSI tags are included in an OPTIONS request, consistently with [RFC3840], IARI tags or ICSI tags shall be concatenated using commas as described in the example below:

+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcse.im,urn%3Aurn-7%3A3gpp-application.ims.iari.rcse.ft"

Table 29: IARI tag concatenation format example

2.6.1.1.3 Future extensions to the mechanism

In addition to the aforementioned additions and to allow:

- A Service Provider (or group of Service Providers) to deploy additional services which can benefit from the RCS discovery mechanism, an additional tag format is defined:
  - +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.mnc<mnc>.mcc<mcc>.<service name>"
  - Valid examples are:
    - +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.mnc001.mcc214.serviceA"
    - +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari rcs.mnc680.mcc310.serviceB"

The service name is decided by the each Service Provider. The only requirement for a Service Provider following this approach is to include these tags in the relevant interoperability agreements with other Service Providers.

- A third-party to deploy an Extension (e.g. through device API as per [PRD-RCC.53]) which can benefit from the RCS discovery mechanism, an additional tag format is defined:
  - +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.ext.<identifier>"
    - Where <identifier> is an identifier (encoded in base64 as per [RFC4648]) uniquely identifying the application. The way the system ensures uniqueness of the value of the identifier is out of scope of this specification. The length of this field shall not exceed 64 bytes.
  - A valid example is:
    - +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.ext.A5TgS99bJloIUIh1209SJ82B21m87S1B87SBqfS871BS8787SBXBA3P45wjp63tk”

NOTE: A Service Provider may deploy Extensions as a third-party.

The use of those Extensions for actual third-party usages requires a full functioning framework, including a secure management of the Extensions on the devices. If this framework is not in place, Clients shall not enable third parties to use this tag.

These IARIs are only exchanged in capability discovery and updates when the Extensions make use of the RCS infrastructure to communicate (as per section 3.12.4.2).

RCS service	Tag
Service Provider specific service	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs .mnc<mnc>.mcc<mcc>.<service name>"
Third-Party specific service	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs .ext.<identifier>"

Table 30: SIP OPTIONS tag proposal for future lines of work

2.6.1.1.4 UI integration optimisations

In addition to the optimizations to minimize the traffic generated by the SIP OPTIONS exchanges when possible, there are two additional optimizations related to the discovery mechanism integration on the UI that should be taken into account:

- The round trip time for a SIP OPTIONS exchange (send and receive response) is expected to range between under 1-2 seconds. Taking this into account, the UI has to be optimized to minimize the impact of this exchange delay.
- When sending the SIP OPTIONS messages to several users (for example during first time registration or when polling), it is recommended to employ a non-aggressive strategy and allow time between each exchange to:
  - Minimize potential network impact
  - Avoid any impact on the user experience (for example a slower UI, blockings and so on)

Please note that in this case this specification does not specify the specific mechanisms which should be implemented leaving space to Original Equipment Manufacturers (OEMs)

and third parties to drive innovative and differentiated solutions, which distinguish their products from competitors.

2.6.1.1.5 Multidevice support: Options-AS

Ultimately, the choice of supporting multiple devices for a single user is decided by each Service Provider. The considerations contained in this section will only apply to those Service Providers willing to include RCS multidevice support in their networks.

In a multidevice scenario, when the user is registered to the IMS CORE with various devices using the same URI (that is the same implicit registration set), the OPTIONS exchange will return incomplete information:

- The capabilities contained in the OPTIONS message refer only to the originating device (that is the originating user may be logged in with the same URI in several devices).
- The IMS Core, depending on the configuration, either sends the OPTIONS message to the device that first registered to the IMS CORE or forks the OPTIONS to all of the registered devices. In any case, only the first response is passed back to the requester, discarding the others. In other words, the capabilities returned in the OPTIONS response will be from only one of the user's devices.

The preferred implementation for handling the OPTIONS in a multidevice environment is left to the Service Provider's discretion. The only requirement is that it should not impact the terminal side (that is there will be no changes on the client side). A possible solution for extending the OPTIONS mechanism to a multidevice scenario is to include a custom AS implementing the following logic:

- A trigger will be setup in the IMS CORE to send all of the OPTIONS from an RCS user to the AS.
- The AS will fork the OPTIONS request to all of the RCS user's registered devices and will aggregate all of the capabilities returned into one OPTIONS response if the forking is not already implemented by the IMS core network.
- Once the responses from the different devices are received, the AS will aggregate all the capabilities from the replies and send them back to the caller.
- Even if not all of the replies have been received in less than a configurable amount of time (note the recommendation is to set the value to optimise the UX on the terminal) the AS will return the aggregated information received so far.
- Capabilities shall be aggregated to provide the response to an incoming SIP OPTIONS request. For outgoing requests, it is up to the Service Provider's policy to aggregate the capabilities.

  NOTE: Similar procedures may at the service provider's discretion also be applied at originating side to aggregate the capabilities of all the user's devices in the OPTIONS request.

To implement this feature, an application server should be able to uniquely identify each user device to perform the forking of the OPTIONS message and to intercept and process the responses. The mechanism to have these individual identities (a GRUU or sip.instance feature tag) is covered in section 2.11.3.

While multidevice support is an option for each Service Provider to decide whether or not it is supported, the RCS capability discovery mechanism based on the SIP OPTIONS message is a mandatory requirement and the behaviour will be the one specified previously to ensure seamless interworking between Service Providers.

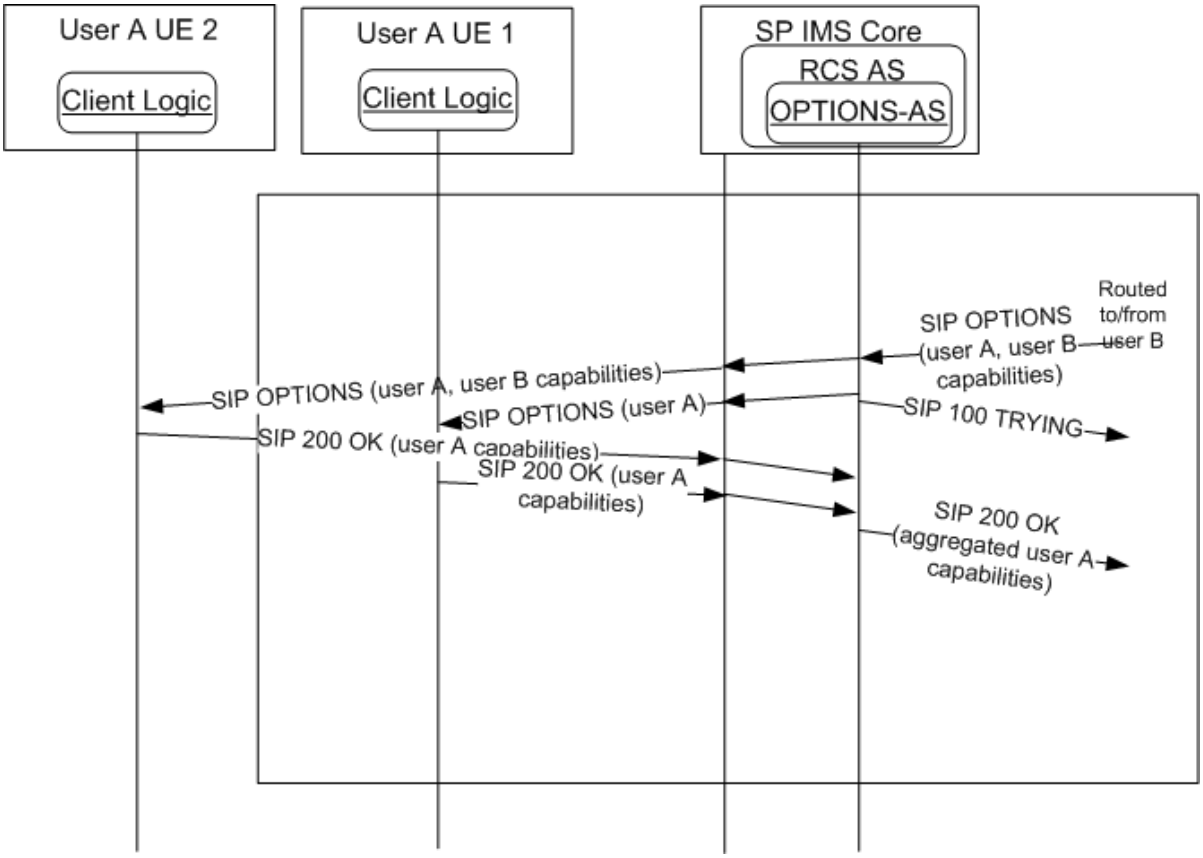


Figure 22: Options application server: Capability aggregation on SIP OPTIONS request

2.6.1.2 Capability discovery via presence

2.6.1.2.1 General Overview

As an alternative to the SIP OPTIONS-based mechanism presented in the previous section, a Service Provider deploying a Presence Server may provide the capability discovery mechanism via presence. The service capabilities are then realized using the "*Service*" part of the Presence Data Model. This part is described in section 2.6.1.2.5.

2.6.1.2.2 Publication of the Service Capabilities

The capabilities are announced in a Presence document that is published by using the SIP PUBLISH method as defined in [Presence]. When the terminal is started, the client then sends a SIP PUBLISH request containing the capabilities (see section 2.6.1.2.5).

The publication is maintained in the Presence Server whenever the application is running by sending a refresh request before it expires.

If changes are required in the published capabilities (for example due to the behaviour specified in sections 2.6.2 and 2.6.3), a presence modify request is sent using the '*Sip-If-Match*' header according to [Presence]. When the client/device is switched off, it shall remove the published capabilities before unregistering according to the procedure defined in [RFC3903] (i.e. by sending a SIP PUBLISH request without a body including the '*Sip-If-Match*' header and an *Expires* header set to 0).

2.6.1.2.3 Service Capabilities Retrieval

Service capabilities of an RCS user can be retrieved by another RCS user via a presence subscription issued by their client, providing the pertaining Presence Authorization rules allow him to do so. The templates provided in sections 2.14.1 and 3.7.4.5.2 allow this for the

authorized users. An RCS user is therefore allowed to retrieve the service capabilities of contacts when they have an established Social Presence relationship.

RCS users may also retrieve the service capability information of contacts with which they have not established a Social Presence relationship by means of Anonymous Fetch operations issued by their client (as described in section 7.1 of [PRESENCE]). This will result in a single NOTIFY request indicating the service capabilities of that contact. This information shall then be cached in the client as described in section 2.6.4. The Anonymous Fetch operation shall be supported in clients.

If an RLS-URI (Resource List Server URI, see Annex A section A.1.1.1) has been provisioned, a client shall use an Anonymous Fetch request using a request-contained list if the client has to query the capabilities of multiple users at once (e.g. during a poll). In this case it shall do so according to section 5.2.1.2.2 of [Presence2.0_TS].

If only a single contact needs to be queried, an individual fetch shall be done instead even if an RLS-URI has been configured.

2.6.1.2.3.1 General Processing Rules to Ensure Backwards Compatibility

To maintain enough flexibility and not to impose potentially sub-optimal technical choices on future RCS versions, the parsing of the capabilities in an RCS client should be sufficiently robust. First the watcher should apply the processing rules defined in [Presence2.0_DDS] and if then there are still multiple elements the watcher shall follow the guidelines in the RCS presence parsing presented below:

- Unknown or unsupported elements and tuples could be present in the document. In that case they should be ignored.
- Unknown service identifiers (Service-Id) could be present in the document. Tuples containing those should be ignored.
- Unknown service versions of known services could be present in the presence document. Tuples containing those should be ignored.
- The same service could occur multiple times in the presence document with different contact addresses. To cope with this case, the following behaviour shall be used for displaying and using the tuples:
    - If one of the tuples contains a contact address that corresponds to the presentity about which the presence document was received, all others shall be ignored.
    - Tuples that contain a contact (address) element which corresponds to another presentity (that is another contact in the contact-list of the user or another tel URI) shall be ignored.
    - Tuples containing contact elements with types of addresses that are not supported by the client for that service shall be ignored (for example messaging using an e-mail address while e-mail is not supported by the client)
    - If after applying the above rules, there are still multiple non-ignored tuples remaining for the service, all but the first shall be ignored.
    - If after applying the above rules there is a non-ignored tuple remaining, the service behaviour shall be as follows
        - The capability to use the service for communication with the contact shall be announced to the user
        - If the remaining tuple contained no contact address or it matched the one of the presentity, the presentity's address will be used for setting up communication using that service

  - Otherwise the address contained in the contact element will be used for setting up the corresponding service
- The Watcher shall follow the procedures defined in section 6.2 "Default Watcher Processing" of [Presence2.0_DDS].

Regarding the use of the address provided in the contact, the communication addresses (contact) part of service tuples shall not be:

- Shown to the end-user, these addresses are handled locally by the terminal;
- Used to request presence subscription, an RCS client is NOT supposed to subscribe to the contact associated with a service capability tuple received in a presence document.

2.6.1.2.4 Authorization for capabilities retrieval

To provide authorization to retrieve the capabilities using an Anonymous Fetch request, an RCS client supporting the capability exchange using presence shall set the presence rules document in the presence XDMS as follows:

Presence XDMS:
AUID: org.openmobilealliance.pres-rules
Document name: pres-rules
Template

```
<?xml version="1.0" encoding="UTF-8"?>
<cr:ruleset
    xmlns:ocp="urn:oma:xml:xdm:common-policy"
    xmlns:op="urn:oma:xml:prs:pres-rules"
    xmlns:pr="urn:ietf:params:xml:ns:pres-rules"
    xmlns:cr="urn:ietf:params:xml:ns:common-policy">

    <! -- This rule allows all service capabilities to be sent for anonymous requests -->
    <! -- To realize the service capabilities to all requirement -->
    <! -- This rule replaces the default “wp_prs_block_anonymous” rule -->
    <! -- NOTE: May be modified to only allow RCS specified services -->
    <cr:rule id="rcs_allow_services_anonymous">
        <cr:conditions>
          <ocp:anonymous-request/>
        </cr:conditions>
        <cr:actions>
          <pr:sub-handling>allow</pr:sub-handling>
        </cr:actions>
        <cr:transformations>
          <pr:provide-services>
              <pr:all-services/>
          </pr:provide-services>
          <pr:provide-all-attributes/>
        </cr:transformations>
    </cr:rule>
</cr:ruleset>
```

Table 31: Presence XDMS template

If social presence is supported (see section 3.7), the *pres-rules* document should be set to contain both “*rcs_allow_services_anonymous*” described in this section and the rules provided in the template described in section 3.7.4.5.2.

Handling of this template shall be done as described in section 2.14.2.

2.6.1.2.5 Service part of the presence Data Model

A service capability is provided according to the model described in Table 32:

Attribute	Specification	Comment
entity	[RFC3863]	The entity field should be populated with a tel URI provided that the device has received a tel URI in P-Associated-URI header of 200 OK response to REGISTER request.
Tuple: <presence> -> <tuple>	[RFC3863] and [Presence2.0_DDS]	According to the presence schema defined in the [Presence], services are presented with *tuple* elements.
Status <tuple> -> <status> -> <basic> -> Open	[RFC3863] and [Presence2.0_DDS]	Mandatory element in [RFC3863]. Once a tuple element is published the value 'open' will always be used. It does not have any particular meaning in RCS context.
Service-id <tuple> -> <service-description> -> <service-id>	[Presence2.0_DDS]	*Service-description* element identifies a service and is described by a *service-id* and *version*. *Service-id* element contains a string that identifies a single service.
Version <tuple> -> <service-description> -> <version>	[Presence2.0_DDS]	*Version* element contains the version number for the service, to identify different versions of the service (for example version number for specification number).
Media <tuple> -> <servcaps>	[RFC5196] and [Presence2.0_DDS]	Indicates the capabilities of the service. In RCS this is only used to provide media capabilities for some specific services (where mentioned below)
Contact <tuple> -> <contact>	[RFC3863] and [Presence2.0_DDS]	Contact element contains Presentity's communication address for the service. Contact address can be for example a tel or SIP URI, depending on the service used. The use of the Contact element is optional (if used it has to be a global routable URI) since the watcher may use the URI stored in the address book when initiating communication with the presentity. RCS Presentities either do not insert any contact element or insert a contact element for which the address matches the one used for identifying itself in communication (see Section 2.5) Note 1: According to [RFC3863], "tuples that contain a <*basic*> element SHOULD contain a <*contact*> address". Note 2: Populating <*contact*> element with GRUU may result in unpredictable watcher behaviour (see section 2.6.1.2.3.1) so GRUU should not be used Therefore -as a default- the <*contact*> element should be populated with a tel URI provided that: • The device has received a tel URI in P-Associated-URI header of 200 OK response to REGISTER request. • The service in question can utilize tel URIs.

Timestamp: <tuple> -> <timestamp>	[RFC3863] and [Presence2.0_DDS]	Timestamp when the presence information was published.

Table 32: Attributes of the Presence Service element

2.6.1.2.5.1 Service-descriptions for the Selected RCS Services

Service capabilities publication through OMA Presence Enabler [Presence2.0_TS] or [Presence] must follow [PDE_13] rules.

The RCS registered Service-description values are listed in OMNA Presence <service-description> Registry at:

http://www.openmobilealliance.org/Tech/omna/omna-prs-PidfSvcDesc-registry.aspx

The following *<service-description>* child elements, will be used in the presence document:

Standalone Messaging
Service-id: *org.openmobilealliance:StandaloneMsg*
Version: 2.0
Contact address type: tel / SIP URI

Session Mode Messaging
Service-id: *org.openmobilealliance:IM-session*
Version: 1.0
Contact address type: tel / SIP URI

Or

Service-id: org.openmobilealliance:ChatSession
Version: 2.0
Contact address type: tel / SIP URI

Full Store and Forward Group Chat
Service-id: *org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.fullsfgroupchat*
Version: 1.0
Contact address type: TEL/ SIP URI

File Transfer
Service-id: *org.openmobilealliance:File-Transfer*
Version: 1.0
Contact address type: tel / SIP URI

File Transfer (with Store and Forward)
Service-id: *org.openmobilealliance:File-Transfer*
Version: 2.0
Contact address type: tel / SIP URI

File Transfer Thumbnail
Service-id: *org.openmobilealliance:File-Transfer-thumb*
Version: 2.0
Contact address type: tel / SIP URI

File Transfer via HTTP
Service-id: *org.openmobilealliance:File-Transfer-HTTP*
Version: 1.0
Contact address type: tel / SIP URI

Image Share
Service-id: org.gsma.imageshare
Version: 1.0
Contact address type: tel / SIP URI

Video Share during a call (Phase 1)
Service-id: *org.gsma.videoshare*
Version: 1.0
Contact address type: tel / SIP URI

Video Share outside of a voice call (Phase 2)
Service-id: *org.gsma.videoshare*
Version: 2.0
Contact address type: tel / SIP URI

Social presence information
Service-id: *org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcse.sp*
Version: 1.0
Contact address type: tel / SIP URI

Capability discovery via presence
Service-id: *org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcse.dp*
Version: 1.0
Contact address type: tel / SIP URI

IP Voice Call (IR.92)
Service-id: *org.3gpp.urn:urn-7:3gpp-service.ims.icsi.mmtel*
Version: 1.0
Media capabilities: audio, duplex
Contact address type: tel / SIP URI

IP Video Call (IR.94)
Service-id: *org.3gpp.urn:urn-7:3gpp-service.ims.icsi.mmtel*
Version: 1.0
Media capabilities: audio, video, duplex
Contact address type: tel/ SIP URI

NOTE1: A single device supporting both IP Voice Call (IR.92) and IP Video Call (IR.94) shall publish a single tuple containing the common MMTel service id and both audio and video servcaps elements. Separate tuples are not required.

RCS IP Voice Call (strong preference for no breakout (IP end to end))
Service-id: *org.3gpp.urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall*
Version: 1.0
Media capabilities: audio, duplex
Contact address type: tel / SIP URI

RCS IP Video Call (strong preference for no breakout (IP end to end))
Service-id: *org.3gpp.urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall*
Version: 1.0
Media capabilities: audio, video, duplex
Contact address type: tel/ SIP URI

NOTE2: A single device supporting both RCS IP Voice Call and RCS IP Video Call with a strong preference for no breakout shall publish a single tuple containing the org.3gpp.urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall service-id and both audio and video servcaps elements. Separate tuples are not required.

RCS IP Video Call (strong preference for no breakout (IP end to end) and video media cannot be dropped)
Service-id: *org.3gpp.urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall.ipvideocallonly*
Version: 1.0
Media capabilities: audio, video, duplex
Contact address type: tel/ SIP URI

NOTE3: A device shall provide in the published capabilities only one of the RCS IP Voice Call, RCS IP Video Call and RCS IP Video Call where video media cannot be removed capabilities depending on whether according to the PROVIDE RCS IP VOICE CALL and PROVIDE RCS IP VIDEO CALL configuration parameters defined in section A.1.14 in the currently used radio technology respectively only RCS IP Voice Calls, both RCS IP Voice Calls and RCS IP Video Calls or only RCS IP Video Calls are allowed. Due the combining of the capabilities of different devices (i.e. the Presence Composition policy) it may happen though that multiple of those capabilities are received by a watcher. If the RCS IP Video Call where video media cannot be removed capability is received in combination with the RCS IP Video Call and/or the RCS IP Voice Call capability, towards that contact both RCS IP Voice Call and RCS IP Video Calls may be successfully initiated as described in sections 3.8 and 3.9 when those services are allowed according to the local configuration and used radio technology.

NOTE4: RCS IP Calls and IP Calls per IR.92/IR.94 can technically interoperate. Whether such interoperability is provided is based on Service Provider decision. In those cases the Service Provider should ensure that all relevant capabilities are provided in the capability exchange (e.g. if IP Voice Call (as per MMTEL) capability is provided, the RCS IP Voice Call capability should be provided also).

Geolocation PUSH
Service-id: *org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.geopush*
Version: 1.0
Contact address type: tel/ SIP URI

Geolocation PULL
Service-id: *org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.geopull*
Version: 1.0
Contact address type: tel/ SIP URI

Geolocation PULL File Transfer
Service-id: *org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.geopullft*
Version: 1.0
Contact address type: tel/ SIP URI

NOTE5: an RCS client shall include both the Video Share 2.0 and the Video Share 1.0 capabilities to indicate backwards compatibility with earlier RCS clients.

The service capability information that is the object of a SIP PUBLISH by the RCS client (service tuple) corresponds to the services supported by the device. For example, a device in RCS-AA mode (see section 2.2.1) can indicate its support for RCS IP Voice Calls according to section 3.8 with a service- and media description.

The set of services published may be further restricted by some Service Provider settings on the User Equipment (UE, on for example the services that are allowed by the Service Provider in the network) that are described in Annex A.

2.6.1.2.6 Future extensions to the mechanism

Consistently with section 2.6.1.1.3, it is also possible to extend the capability discovery based in presence following the guideline presented in the table below to define new service-IDs:

RCS service	Tag
Service Provider specific service (based on IARI)	Service-Id: org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.mnc<mnc>.mcc<mcc>.<service name> Version: Service Provider choice
Service Provider specific service (based on OMA scheme)	Service-Id: org.openmobilealliance:<RCS service name>.mnc<mnc>.mcc<mcc>.<service extension> Version: Service Provider choice
Third-Party specific service	Service-Id: org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.ext.<identifier>"

Table 33: Presence service tuple proposal for future lines of work

Service extension patterns including “mnc.mcc” may be registered with OMNA, if a service provider wishes to reserve the values in order to avoid any future collisions with new services (extensions, or new OMA services).

Example of the reserved service extension patterns that may be registered is:

org.openmobilealliance:ChatSession.mnc072.mcc01

Examples of service extensions:

- Service-id extension(s) for Group Chat with full store and forward:
  org.openmobilealliance:ChatSession.mnc072.mcc01
  OR
  org.openmobilealliance:ChatSession.mnc072.mcc01.myGCFlavor1 AND
  org.openmobilealliance:ChatSession.mnc072.mcc01.myGCFlavor2
- Using IARI:

  org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.mnc01.mcc072.fullsfgroupchatMyFlavor

The use of the Third-Party specific services requires a full functioning framework, including a secure management of the Extensions on the devices. If this framework is not in place, Clients shall not enable third parties to use this service.

The Third-Party specific services are only exchanged in capability discovery and updates when the Extensions make use of the RCS infrastructure to communicate.

2.6.1.3 Coexistence between the discovery mechanisms

2.6.1.3.1 Service/capability indicators

The equivalence between presence Service-IDs and SIP OPTIONS tags are presented in the following table:

RCS service		Tag
Standalone Messaging	Tag	+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.oma.cpm.msg,urn%3Aurn-7%3A3gpp-service.ims.icsi.oma.cpm.largemsg"

	Service Tuple	Service-id: org.openmobilealliance:StandaloneMsg Version: 2.0
Chat	Tag	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcse.im"
	Service Tuple	Service-id: org.openmobilealliance:IM-session Version : 1.0 Or Service-id: org.openmobilealliance:ChatSession Version : 2.0
Full Store and Forward Group Chat	Tag	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.fullsfgroupchat"
	Service Tuple	Service-Id: org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.fullsfgroupchat Version: 1.0
File Transfer	Tag	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcse.ft"
	Service Tuple	Service-id: org.openmobilealliance:File-Transfer Version : 1.0
File Transfer Thumbnail	Tag	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.ftthumb"
	Service Tuple	Service-id: org.openmobilealliance:File-Transfer-thumb Version : 2.0
File Transfer Store and Forward	Tag	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.ftstandfw"
	Service Tuple	Service-id: org.openmobilealliance:File-Transfer Version : 2.0
File Transfer via HTTP	Tag	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.fthttp"
	Service Tuple	Service-id: org.openmobilealliance:File-Transfer-HTTP Version : 1.0
Image Share	Tag	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.gsma-is"
	Service Tuple	Service-id: org.gsma.imageshare Version: 1.0
Video Share during a call	Tag	+g.3gpp.cs-voice
	Service Tuple	Service-id: org.gsma.videoshare Version: 1.0
Video Share outside of a voice call	Tag	+g.3gpp.iari-ref=”urn:urn-7:3gpp-application.ims.iari.gsma-vs"
	Service Tuple	Service-id: org.gsma.videoshare Version: 2.0
Social presence information	Tag	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcse.sp"
	Service Tuple	Service-id: org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcse.sp Version: 1.0

Capability discovery via presence	Tag	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcse.dp"
	Service Tuple	Service-id: org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcse.dp Version: 1.0
IP voice call (IR.92/IR.58)	Tag	+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel"
	Service Tuple	Service-id: org.3gpp.urn:urn-7:3gpp-service.ims.icsi.mmtel Version: 1.0 Media capabilities: audio, duplex
RCS IP voice call	Tag	+g.gsma.rcs.ipcall
	Service Tuple	Service-id: org.3gpp.urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall Version: 1.0 Media capabilities: audio, duplex
IP video call (IR.94)	Tag	+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel";video
	Service Tuple	Service-id: org.3gpp.urn:urn-7:3gpp-service.ims.icsi.mmtel Version: 1.0 Media capabilities: audio, video, duplex
RCS IP video call	Tag	+g.gsma.rcs.ipcall;video
	Service Tuple	Service-id: org.3gpp.urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall Version: 1.0 Media capabilities: audio, video, duplex
	Tag (RCS IP video call where video media cannot be removed)	+g.gsma.rcs.ipvideocallonly
	Service Tuple (RCS IP video call where video media cannot be removed)	Service-id: org.3gpp.urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall.ipvideocallonly Version: 1.0 Media capabilities: audio, video, duplex
Geolocation PULL	Tag	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.geopull"
	Service Tuple	Service-id: org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.geopull Version: 1.0
Geolocation PULL using File Transfer	Tag	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.geopullft"
	Service Tuple	Service-id: org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.geopullft Version: 1.0
Geolocation PUSH	Tag	+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.geopush"

	Service Tuple	Service-id: org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.geopush Version: 1.0

Table 34: Complete SIP OPTIONS tag and Presence Service ID usage for RCS

2.6.1.3.2 Coexistence using a common device stack

As mentioned in section 2.6.1, the principle for interoperability is to have a common stack on devices which is able to:

- Answer a SIP OPTIONS query as per the mechanism presented in section 2.6.1.1 independently on whether the device is configured to use SIP OPTIONS or presence as the default capability exchange mechanism.
- If the device is configured to use presence as the default capability exchange mechanism, implement the fallback to SIP OPTIONS procedure

2.6.1.3.2.1 Interworking when the request is originated in the Service Provider using presence as the default discovery mechanism

In this case, the initial capability exchange request is performed using presence (ANONYMOUS SUBSCRIBE), however either the originating or the terminating Service Provider Network detects that this method is not supported for that particular user and returns with one of the following errors:

- 405 METHOD NOT ALLOWED
- 501 NOT IMPLEMENTED

As a result, the RCS stack on the UE shall identify that the contact does not support the Presence discovery mechanism. From there on the OPTIONS-based mechanism (as presented in section 2.6.1.1) shall be used to query that contact's capabilities.

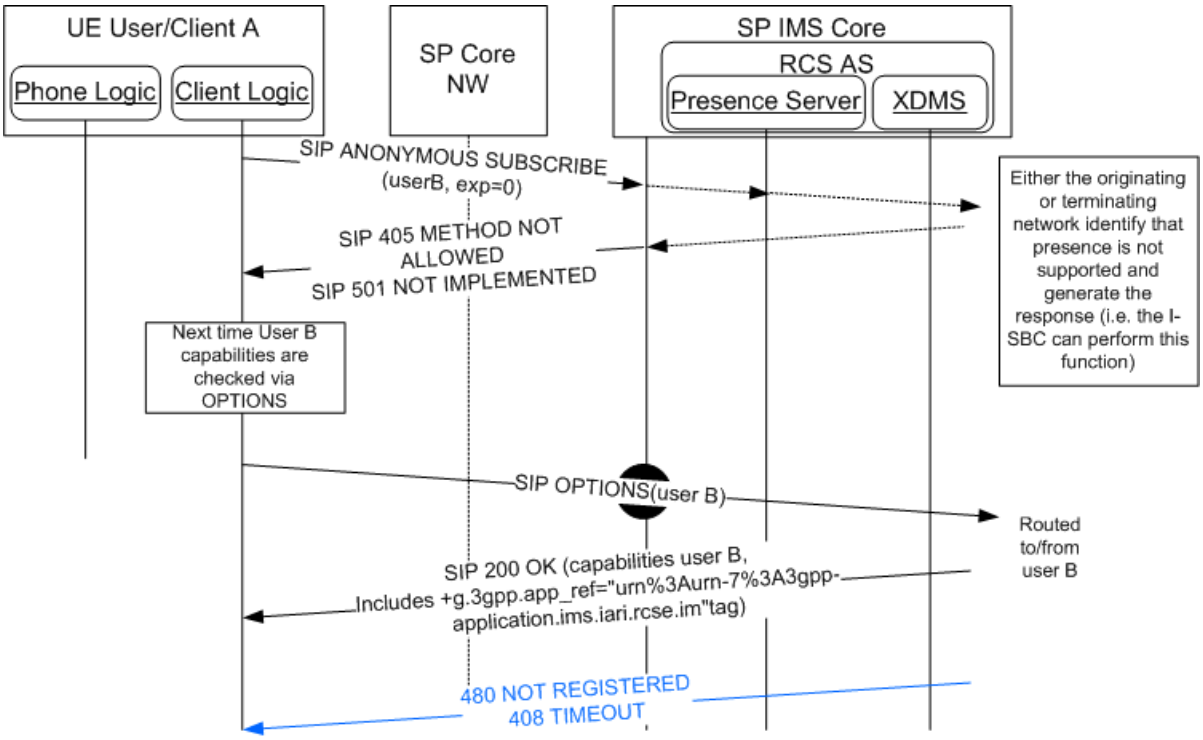


Figure 23: Fallback to SIP-OPTIONS procedure

If in the future, the contact is again identified as supporting discovery via presence (i.e. the *+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcse.dp"* tag was included

either in the OPTIONS request or in its response), then capability discovery via presence (as described in section 2.6.1.2) will be used from there on for that contact.

2.6.1.3.2.2 Interworking when the request is originated in the Service Provider using SIP OPTIONS as the default discovery mechanism

In this case, the SIP OPTIONS message is exchanged end-to-end as described in section 2.6.1.1.1.

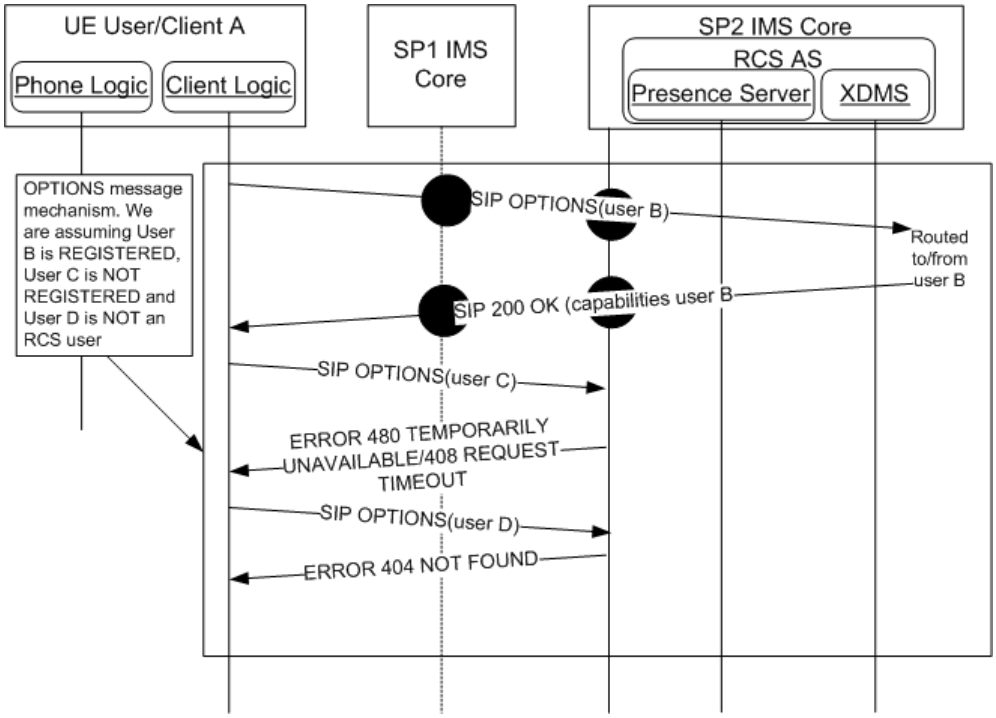


Figure 24: Inter-Service Provider SIP OPTIONS exchange for interworking

2.6.1.3.3 Coexistence between the discovery mechanisms via network interworking

When Service Providers use presence as the default discovery mechanism, there are two ways in which interoperability is achieved between such a Service Provider and those Service Providers who have selected SIP OPTIONS as the default discovery mechanism.

- Service Provider supports fallback to SIP OPTIONS: Interoperability leverages the common device stack as defined in 2.6.1.3.2 above. In this case, there is no requirement for network based interworking
- Service Provider does not support fallback to SIP OPTIONS: Interoperability is provided by network based interworking. Refer to the interworking table below to identify specific network based interworking requirements:

			Service Provider A			
			Default: SIP OPTIONS		Default: Presence	
			No Presence Server	Presence Server	OPTIONS Fallback	No OPTIONS fallback
Service Provider B	Default: SIP OPTIONS	No Presence Server	No Network Interworking Required[2]	No Network Interworking Required[2]	No Network Interworking Required[1]	**Bidirectional Interworking required[3]**
		Presence Server	No Network Interworking Required[2]	No Network Interworking Required[2]	No Network Interworking Required[1]	**Unidirectional Interworking required[4]**
	Default: Presence	OPTIONS Fallback	No Network Interworking Required[1]	No Network Interworking Required[1]	No Network Interworking Required[2]	No Network Interworking Required[2]
		No OPTIONS fallback	**Bidirectional Interworking required[3]**	**Unidirectional Interworking required[4]**	No Network Interworking Required[2]	No Network Interworking Required[2]

Table 35: Service Discovery network-based Interworking summary

Notes:

1. No interworking required; based on common stack approach
2. No interworking required; based on common default discovery mode
3. Interworking required for SIP OPTIONS conversion to SUBSCRIBE/NOTIFY and vice versa
4. Interworking required for SIP OPTIONS conversion to SUBSCRIBE/NOTIFY; requirement for conversion from SUBSCRIBE/NOTIFY to SIP OPTIONS contingent upon "SIP OPTIONS default" Service Provider support for Anonymous Fetch at PS

Note that Table 35 considers whether a service provider that uses SIP OPTIONS as the default discovery also supports presence or not:

- The Presence Server acts as a source for both SPI and capability information. This is addressed in 2.6.1.4 which states that capability exchanges are not required in the case where a social relationship is established.
- If a Service Provider uses SIP OPTIONS as the default discovery mechanism, and has deployed presence, the Service Provider may implement a policy that allows their Presence Server to respond to presence based discovery (anonymous) requests.
- Such a policy would impact the required interworking architecture; therefore it is addressed in Table 35 above.

Specific network interworking function requirements are contingent upon the service discovery modes and policies of each service provider. At the Service Provider's discretion, an interworking function can be implemented in the network to:

- Answer incoming SIP OPTIONS requests based on the Presence Server information (Figure 25).
- Convert SIP ANONYMOUS SUBSCRIBE requests into SIP OPTIONS requests (Figure 26).

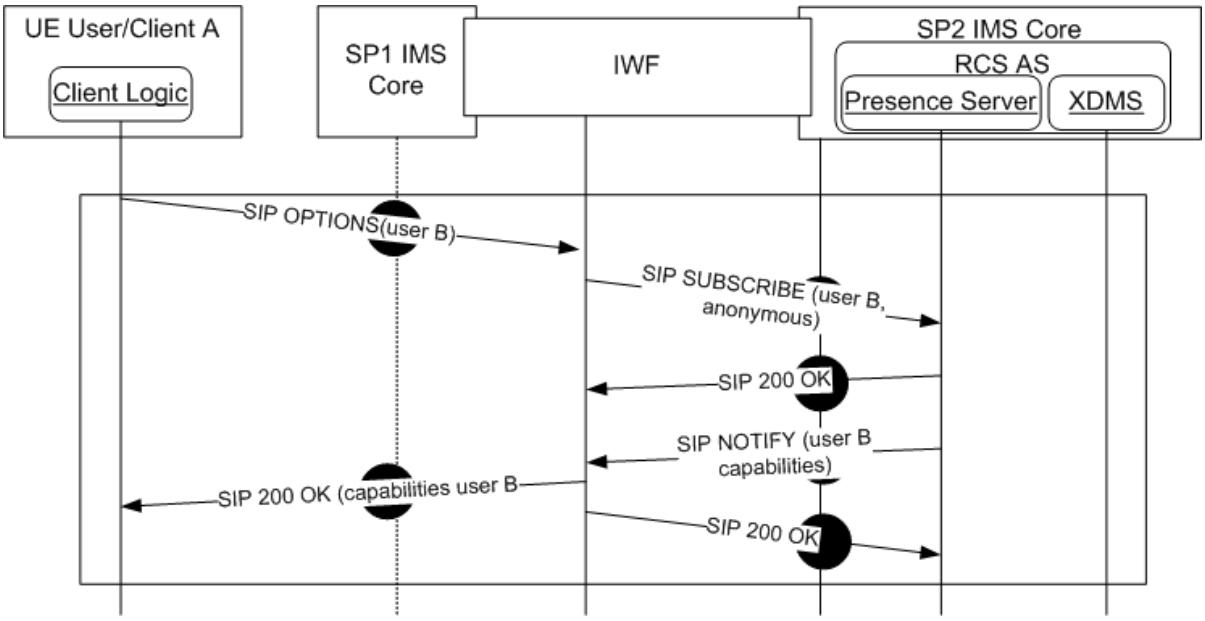


Figure 25: Capability interworking via network: Options request

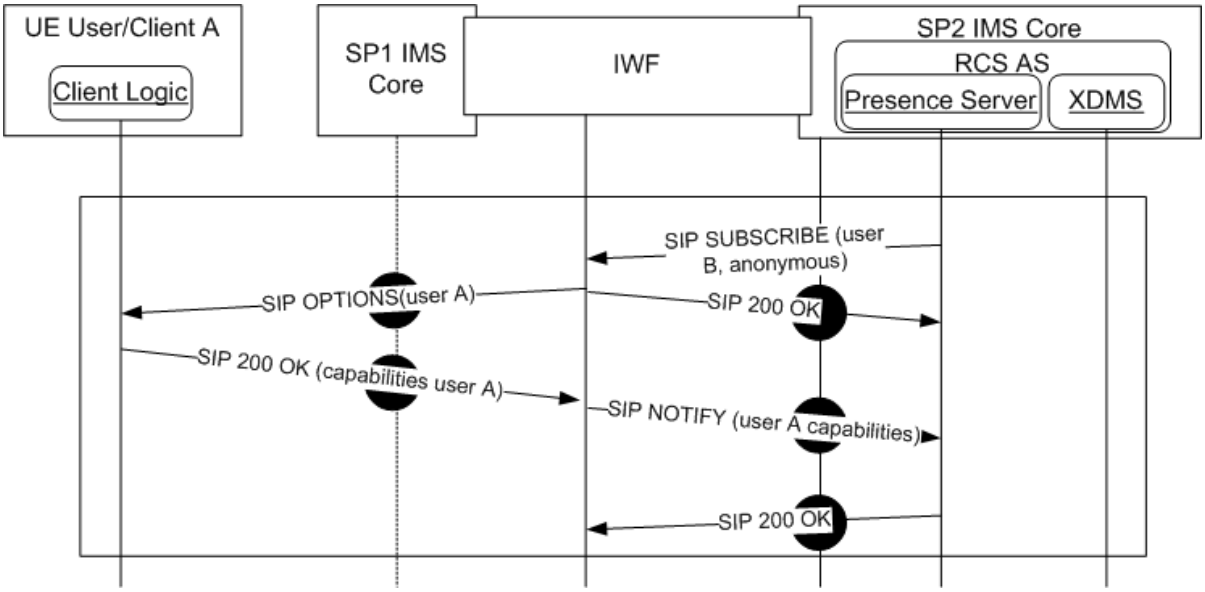


Figure 26: Capability interworking via network: Presence request

Note that Figure 25 and Figure 26 do not specify whether the IWF is deployed:

- In the Originating IMS network
- In the Terminating IMS network
- In the Inter-Network region

All of the above are valid architectural options. NNI impact is not uniform and is a function of the architecture selected. While the details surrounding the specific architecture and functionality of an IWF are left to the Service Provider, it is recommended that impact at the UNI should be minimal and as transparent as possible.

The successful deployment of network IWF capabilities must provide an environment where all RCS devices exchange capabilities information without requiring additional functionality or logic at the client (i.e. no UNI impact).

The following additional guidelines are provided regarding the implementation of an IWF function:

- If either Service Provider has a heterogeneous network from a capabilities discovery mode perspective, this must be factored into the IWF architecture.
- The Service Provider implementing an IWF must consider policy aspects of the functionality. This includes any decisions to filter or transform service capabilities across the IWF.
    - Domain/Service Provider based policies; i.e. specific services are configured to be exposed based on the destination domain.
    - Service level policies: specific services, including Service Provider proprietary or other specialized services that may be filtered from exposure to any external domains
    - User based policy; including privacy or other subscriber level policies

2.6.1.4 Capability discovery and social presence information coexistence

In the following two cases:

- The default mechanism for capability discovery is performed via SIP OPTIONS and the Service Provider has decided to deploy a Presence Server to provide the SPI service[6].
- The default discovery mechanism is based on presence

Then for those contacts who have a social presence relationship established with the sender, it is not necessary to perform a capability exchange because their capabilities will be updated automatically using the standard SPI mechanisms described in section 3.7.4.

2.6.1.5 Capability exchange optimisations

To avoid the overhead and increase the efficiency, the client may implement optimisation mechanisms as listed in section 2.6.4.

2.6.2 User discovery mechanism

With the main aim of optimising the UX and minimising the unnecessary traffic generated by an RCS client, a set of lists shall be generated and maintained by the UE or client:

- A list of the RCS enabled users as the list of users which support at least one RCS service and obviously the capability discovery framework. It should be noted that, the first view of the address book shall use this list to clearly identify the RCS capable contacts with a visual RCS flag.
- One individual list per RCS service of RCS contacts which are enabled to perform that particular service.

These lists should include both registered and non-registered contacts; in contrast, it does not include non-provisioned contacts.

To keep these lists up-to-date, the UE or client shall use one of the capability discovery mechanisms presented in section 2.6.1 in the following scenarios:

- When a new contact is added to the phonebook. The new contact may come from different sources and, therefore, the mechanism described in the following sections applies to all the scenarios presented below:
  - Contact added manually by the user
  - Synchronized via 3rd party servers or PC
  - Received via Bluetooth or handling a vCard file received, for example via e-mail
- The first time the user accesses the service from a new device, the whole address book needs to be polled.
- Periodically (frequency determined by the POLLING PERIOD parameter described in Annex A section A.1.10) to all the contacts in the phone address book whose capabilities are not available (e.g. non-RCS users) or are expired (see CAPABILITY INFO EXPIRY parameter in Annex A section A.1.10 for reference),
- When a contact's details are edited thereby modifying the information which is used to identify the contact as RCS (as described further in section 2.6.2 e.g. the MSISDN is modified or a new MSISDN is added).

[6] It may be possible for a Service Provider to always perform service discovery via SIP OPTIONS, but have a policy allowing for remote domain (NNI) support for discovery via presence as discussed in 2.6.1.3.3. This would allow a remote Service Provider that does not support fallback to SIP OPTIONS to obtain capability information using anonymous SUBSCRIBE without traversing a network IWF.

Additionally, it should be noted that if a client is NOT registered at the time the new contact(s) are added, the client should keep the necessary information on the device. In this case the capabilities shall be verified the next time the RCS client completes the registration process.

2.6.2.1 Discovery via OPTIONS message

The SIP OPTIONS message can be employed not only to determine the capabilities but also to identify whether or not a contact is an RCS user; independently from whether the contact is registered at the time the query is performed.

When a SIP OPTIONS message is sent from User A to User B, User A will learn about user B's capabilities through one of 6 scenarios:

1. If User B is registered and User A is not in the capability blacklist of User B, then the response from User B's client will include the CAPABILITY STATUS – the set of services currently available (based on tags as described in section 2.6.1.1.2), else if User B is registered and User A is in its blacklist, it will only answer User A with an empty 200 OK. Please note that regarding the list of RCS users, the contact shall be only considered as an RCS user, if the response (SIP 200 OK) includes any of the tags described in Table 34.
2. If User B is currently not registered (e.g. the device is switched off, out of coverage or roaming with data services disabled), then the network will respond with one of the following error messages: SIP 480 TEMPORARILY UNAVAILABLE (graceful deregistration took place) or SIP 408 REQUEST TIMEOUT. From the new user discovery point of view, this response is ignored because it is inconclusive:
   - It does not confirm whether the contact is an RCS user, and,
   - It does not provide any relevant update to the list of RCS contacts capable of a particular service
3. If User B is not provisioned for RCS the network will respond with a message error: SIP 404 NOT FOUND[7]. Therefore, if this message is received, the user is identified as a non-RCS user (removed from the list of RCS users and from the individual list of RCS users capable of a particular service)
4. If User B was previously identified as an RCS user and the response to the OPTIONS message indicates that User B is no longer supporting any RCS services, User B should be identified as a non-RCS user and, consequently, removed from the list of RCS enabled contacts
5. In addition to this and based on the fact the SIP OPTIONS request contain the list of services supported by the requester, the receiver shall use the SIP OPTIONS message to update both the RCS contact list and the relevant per service lists as per the criteria presented in the previous four scenarios .
6. Please note there is a possibility an RCS user who is not within the address book contacts may send OPTIONS messages or responses (e.g. when receiving a call or making a call using a MSISDN not included in the contacts). In this case the capabilities

[7] Please note that the response provided may depend on the network configuration. A useful approach for the terminal is to parse the response and if it is not either a 200 OK containing the capabilities as feature tags, a 480 TEMPORARILY UNAVAILABLE or a 408 REQUEST TIMEOUT, the target user should be considered as non-RCS. For simplicity, the present document assumes in the following sections that the response provided by the Service Provider core network is always 404 NOT FOUND, however, the previous statement should be taken into account.

shall be stored temporarily (at least 20 seconds from when OPTIONS is received) in the terminal to:

- Keep the service availability updated while a session (Chat, File Transfer, Video/Image Share, IP Voice or Video call, Geolocation PUSH) is still in place, and,
- To add the information to the new contact (both the fact that it is an RCS user and the cached capabilities) if the user decides to add a new address book entry following a communication.

To illustrate the behaviour, the following example is provided. User A is registered and decides to add or modify a new contact which results in a new IMS identity for the contact (e.g. new MSISDN which implies a new tel URI). As a consequence, the client is required to verify whether the contact is an RCS user and, therefore, add them to the list the terminal maintains.

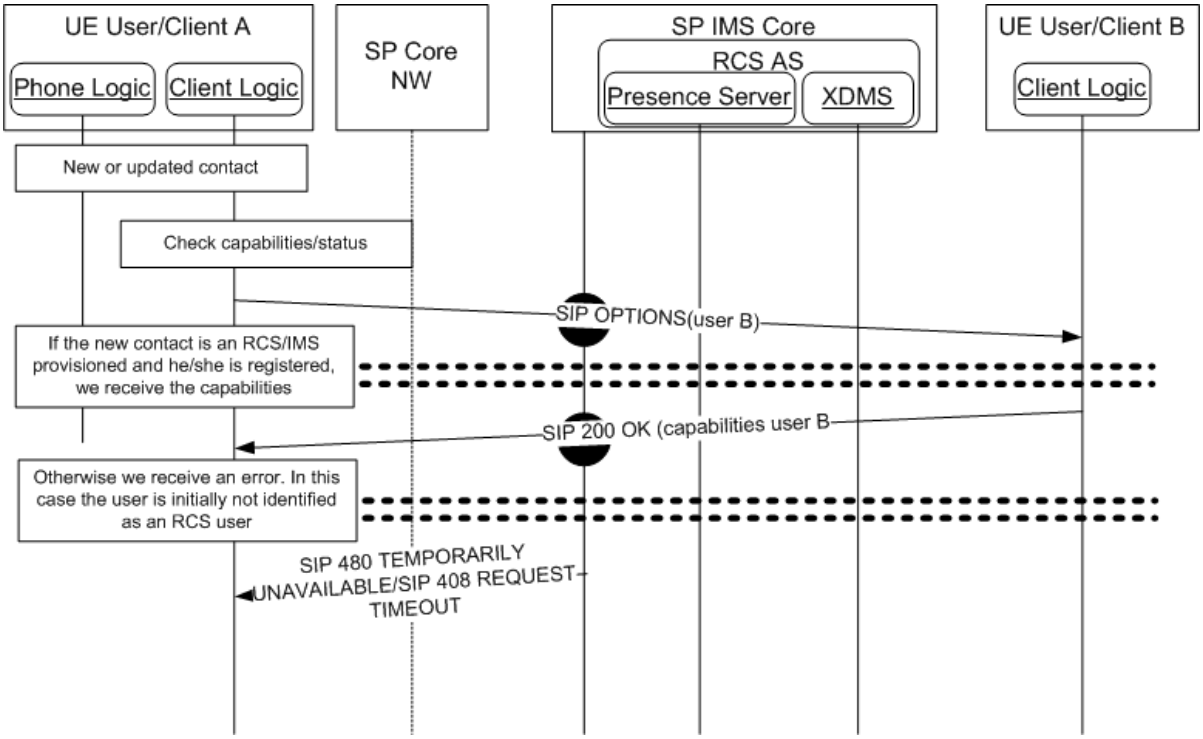


Figure 27: Adding/Editing a contact

2.6.2.2 Discovery via PRESENCE

The procedure for user discovery using presence is analogous to the capability discovery procedure using presence as described in section 2.6.1.2. However the following additional considerations shall be taken into account:

- When User A queries User B's capabilities, the response will include the CAPABILITY STATUS – the set of services currently available (based on the service-IDs presented in section 2.6.1.3.1). Please note that regarding the list of RCS users, the contact shall be considered as an RCS user, only if the response includes one of the service-IDs described in Table 34.

2.6.2.3 Coexistence between user discovery mechanisms

Please note that the mechanisms described in sections 2.6.1.3 also apply to the user discovery mechanisms co-existence.

2.6.2.4 User discovery and social presence information coexistence

Please note that the considerations presented in section 2.6.1.4 also apply to the user discovery process.

2.6.2.5 Capability polling mechanism

To enhance the discovery of new users and, ultimately, keep the list of RCS contacts up to date, this specification provides a mechanism, capability polling, consisting of the polling of the status/capabilities of all the contacts in the address book whose capabilities are not available (such as non-RCS users) or have expired (see CAPABILITY INFO EXPIRY parameter in Annex A section A.1.10 for further reference).

It should be noted that the capability polling mechanism is optional and will be only performed if the related configuration settings have been provisioned (that is if the POLLING PERIOD parameter presented in Annex A section A.1.10 is set to 0, this polling mechanism will not be used).

Assuming the POLLING PERIOD is configured to be greater than 0 and after the polling timer expires, the client will use the following mechanism to update the list of RCS contacts and update their capabilities.

Please note the capability polling is only performed on:

- Those contacts without capability information (non-RCS users and RCS users with unknown capabilities), and,
- The rest of RCS contacts provided the associated capability information is older than the CAPABILITY INFO EXPIRY parameter (see Annex A section A.1.10 for further reference)[8].

[8] Please note this is a traffic optimization to reduce the amount of SIP OPTIONS messages generated by capability polling

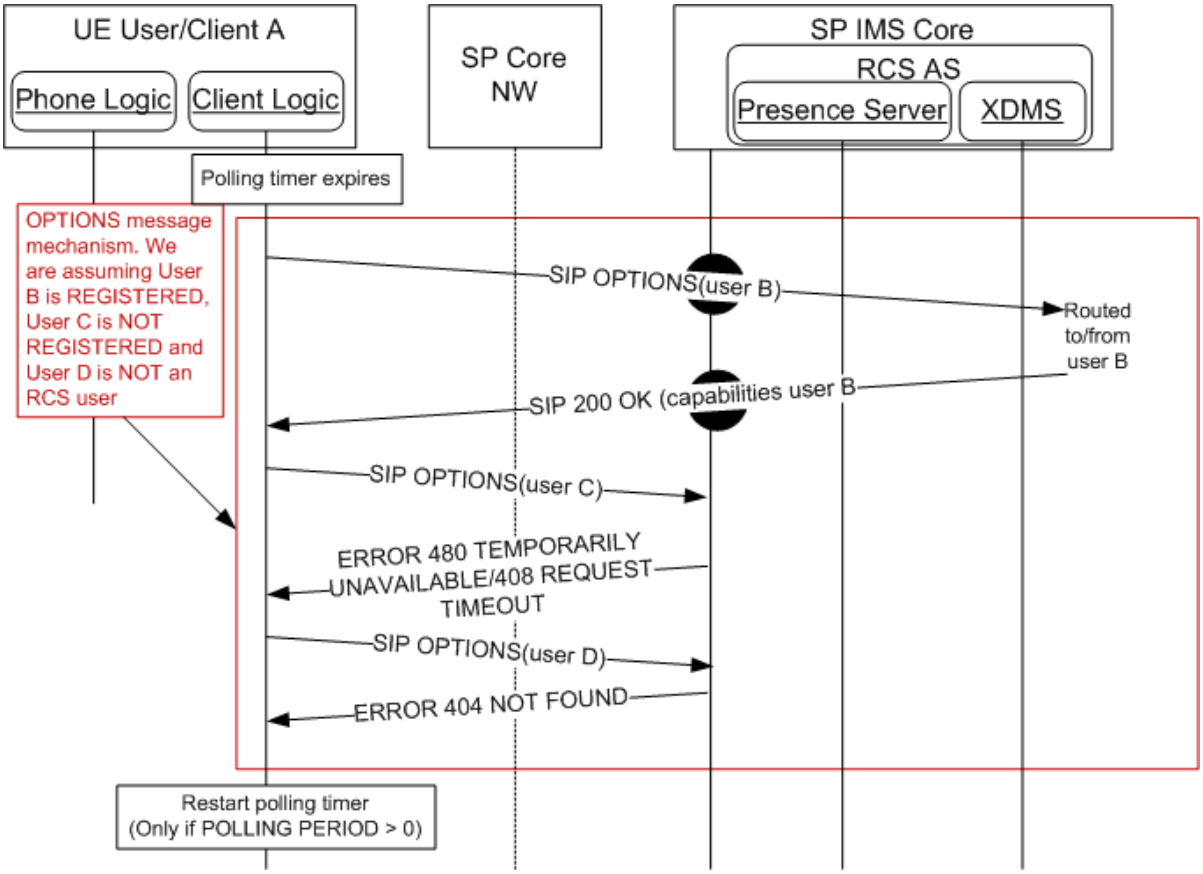


Figure 28: Capabilities polling via OPTIONS message

When CAPABILITY DISCOVERY MECHANISM is set to presence (see Annex A section A.1.10), the presence based discovery based in the use of SIP ANONYMOUS SUBSCRIBE requests are used for all the contacts except:

- If implementing co-existence based on a common device stack, those contacts which are identified as not supporting presence discovery (SIP OPTIONS will be used instead as per the fallback procedure presented in section 2.6.1.3.2.1).
- Those users with a SPI relationship in place because their capabilities will be updated automatically using the standard SPI mechanisms described in section 3.7.4.

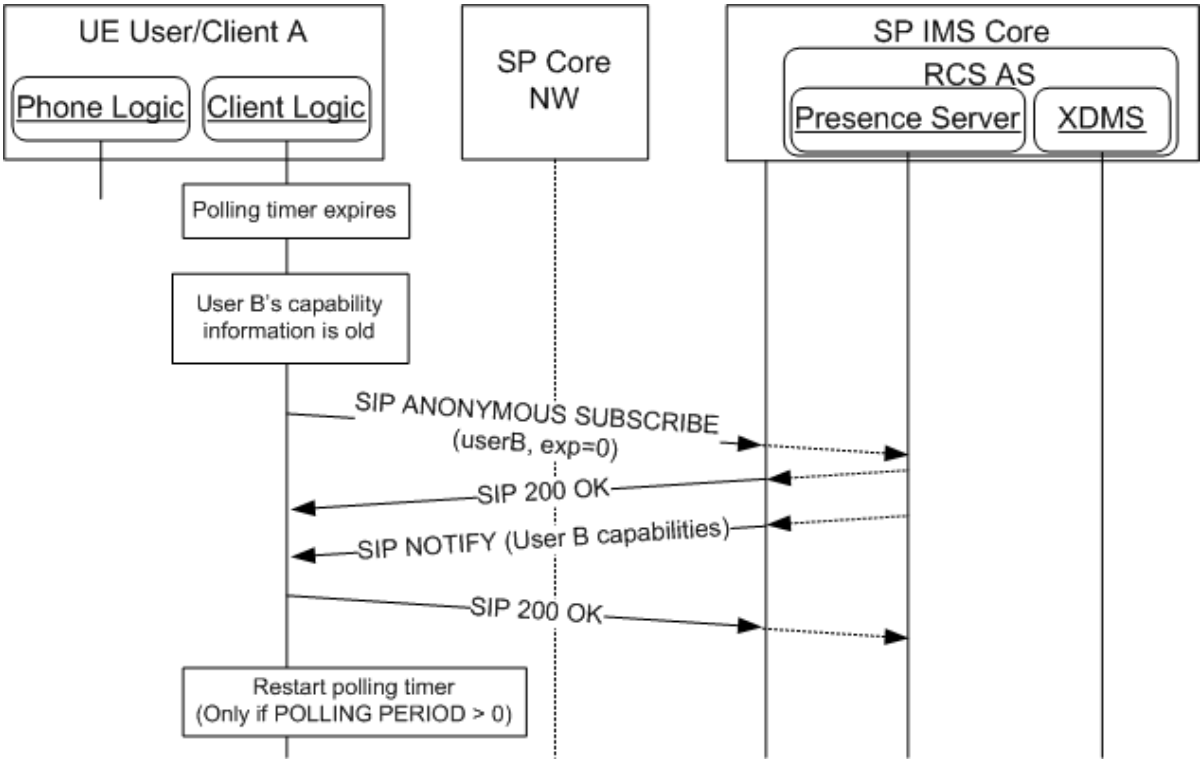


Figure 29: Capabilities polling via anonymous fetch

Note that if an RLS-URI was provisioned (see Annex A Section A.1.1.1) and the capabilities of multiple contacts need to be queried, the capability query could be initiated by the device using a request contained list that is decomposed by the RLS service in the originating network (see section 2.6.1.2.3 for more details). In this case the SIP SUBSCRIBE request shown in Figure 29 would be a back end subscribe issued by the user's home RLS and should be forwarded to the correct destination Presence Server(s). The RLS will gather the notifications and send aggregated notifications to the device.

Finally, and as a summary of the capability and new user discovery mechanism composition the following diagram is provided.

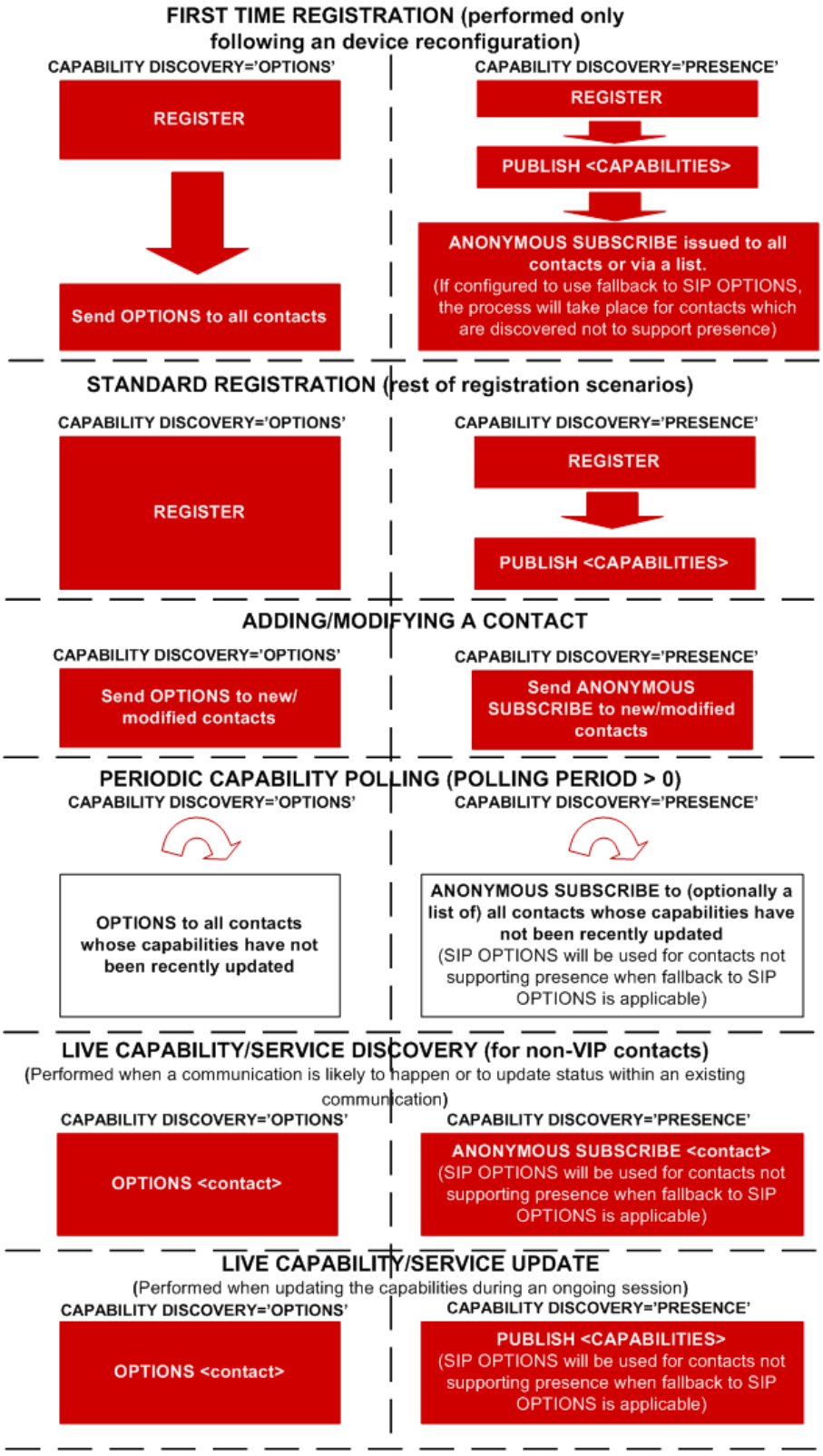


Figure 30: RCS capability and new user discovery mechanisms

The red boxes represent mandatory procedures. Meanwhile the clear boxes represent optional procedures.

2.6.3 Capability update for services

A capability update shall only be triggered for contacts belonging to

- the list of the RCS enabled users as defined in section 2.6.2, or
- the non RCS contacts whose last capability check is older than the NON RCS CAPABILITY INFO EXPIRY configuration parameter value defined in section A.1.10.

2.6.3.1 Entry points for capability update

A capability update is triggered by one of the following activities:

- After first time registration to obtain the registration state and default set of capabilities for each contact in the device's address book (note one capability exchange takes place per IMS identity [that is tel URI/MSISDN or SIP URI] stored in the address book)[9],
- When checking the available RCS services/capabilities to communicate with another user (e.g. from the address book and call-log)
- After establishing voice call to obtain the real-time capabilities for the call or Chat session provided this has not been performed before (see previous bullet) or content sharing during a call is supported.
- After the call returns to an active state (e.g. returning from call wait, call on hold or multiparty call).
- When a communication is active with a user to provide an update when the relevant available capabilities change:
    - When a 1-2-1 Chat session is established and any of the following capabilities change:
        - File Transfer
        - Geolocation PUSH
        - Video Share without a call
    - When in an active call with an RCS user and any of the following capabilities change:
        - Chat
        - File Transfer
        - Geolocation PUSH
        - Geolocation PULL
        - Video Share
        - Image Share
        - IP Video Call
    - When an IP call or video call session is in place and any of the following capabilities change:
        - Chat

[9] Please note a contact may have several MSISDNs or associated SIP URIs. The client will use ALL the MSISDNs/SIP URIs stored for that user to perform the capability exchange. If it is discovered that more than one of the associated tel URIs/SIP URIs are IMS provisioned, each will be treated as a separate RCS user. For example, if displaying the list of RCS contacts, two or more entries for a user will be shown ("John Smith mobile" and "John Smith home"), so the user can choose.

  - File Transfer
  - Geolocation PUSH
  - Geolocation PULL
  - Video Share
  - Image Share
  - IP Video Call
- When there is a communications event (text, email, call or Chat) with another user in the address book, taking into account the optimizations presented in section 2.6.1.5.

2.6.3.1.1 UX guidelines: Access to RCS services through address book and call-log interaction

The address book (and by extension the call-log window as an alternative for users who have been recently phoned) is the centrepiece to access all RCS services. From it, the user is able to:

- Identify which services are available for each contact: When a contact is selected, the capabilities are updated using one of the mechanisms described in section 2.6 (SIP OPTIONS query or PRESENCE), and the result is presented to the user by showing the RCS services which are available to communicate with that particular contact
  - Please note for those contacts who have a social presence relationship established with the sender, it is not necessary to perform a capability exchange because their capabilities will be updated automatically using the standard SPI mechanisms described in section 3.7.4. Therefore and for those contacts, the capability exchange is not required
- If one or more RCS services are available[10], they can be started from the address book/call log entry. Please note the only exception is for those content sharing services that can be only accessed when during a call.
  If a contact has more than one RCS capable telephone number assigned a device should either display for each of these individual numbers which services are available or for each RCS Service the individual telephone numbers on which it is available.

In addition to this, the first view of the address book may clearly identify the RCS capable contacts with an icon or flag.

2.6.3.1.1.1 General assumptions

The following sections describe the relevant chat message flows and reference UX. Please note that the following assumption has been made:

- For simplicity, the internal mobile network interactions are omitted in the diagrams shown in the following sections.

2.6.3.1.1.2 Capability update process

The capabilities update process is described in the following diagram. In this case the contact (User B) is an RCS contact which is registered.

[10] It should be noted that in this case if IM CAP ALWAYS ON (see Table 85) is enabled, the Chat should still be reported to the user as available even if the other end/user is not registered. It may also be offered if the client is not registered itself. In that case the composed messages should be queued to be sent when the device comes online again. The messages should be sent as described in section 2.7.1.1.

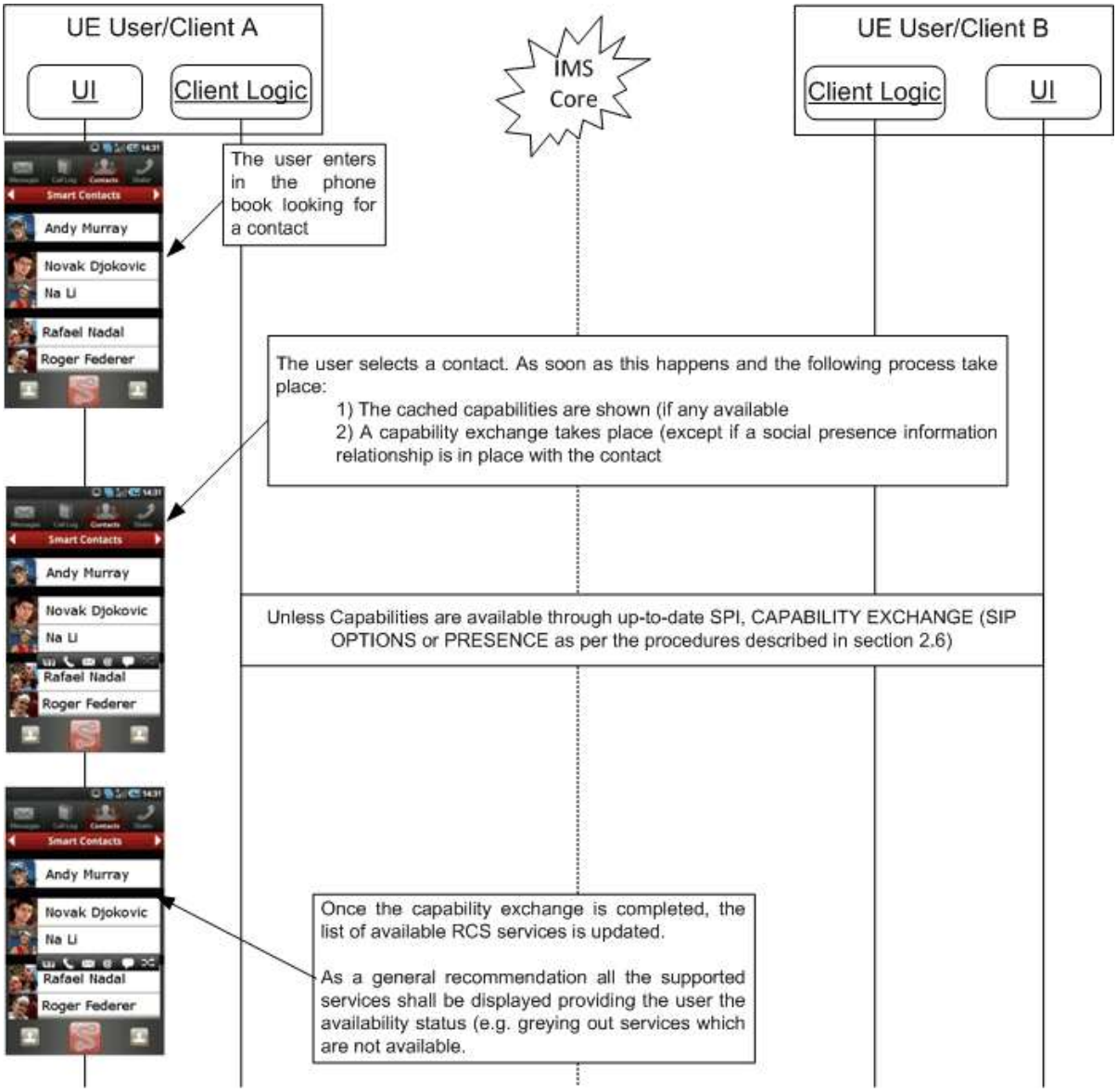


Figure 31 : Address book and call-log service access: Capabilities update

NOTE: If User B is either not an RCS user or they are not currently registered, User A's client may assume that no services are available to communicate with User B.

As a general recommendation all the supported RCS services should be displayed providing the user the availability status (e.g. greying out services which are not available).

2.6.3.2 Standalone messaging: Text and multimedia messaging

The capability exchange is not required for this service.

2.6.3.3 1-to-1 Chat

In addition to the general user driven entry points and taking into account:

- The optimizations provided in section 2.6.1.5, and,
- The potential impact of having an SPI relationship between sender and receiver as described in section 2.6.2.4.

The capability exchange shall take place (when the service capability query is supported by SIP OPTIONS or the contact is not a VIP Contact for SPI):

- Before the initial SIP INVITE is sent to initiate the service to verify if the receiver is ready for the service (unless an exchange of capabilities has just been made based on one of the criteria listed in section 2.6.3.1, as described in section 2.6.4)
- If an error takes place:
  - After the Chat session is abruptly terminated or irregular signalling behaviour during the establishment of the service is detected
  - When there is an update on the available capabilities on either end once the session is established
- In any of the scenarios described in section 2.6.3.1 which are relevant to the service.

2.6.3.4 Group Chat

In addition to the general user driven entry points and taking into account:

- The optimizations provided in section 2.6.1.5, and,
- The potential impact of having an SPI relationship between sender and receiver as described in section 2.6.2.4.

The capability exchange shall take place (when the service capability query is supported by SIP OPTIONS or the contact is not a VIP Contact for SPI):

- Before the initial SIP INVITE is sent to initiate the service to verify if the receiver is ready for the service (unless an exchange of capabilities has just been made based on one of the criteria listed in section 2.6.3.1, as described in section 2.6.4)
- If an error takes place:
  - After the Chat session is abruptly terminated or irregular signalling behaviour during the establishment of the service is detected
  - When selecting the participants of a Group Chat to verify whether they are available

NOTE: UEs may be configured with the full Store and Forward feature using the configuration parameter GROUP_CHAT_FULL_STORE_FWD (see Table 85 in Annex A), with the default value being “false”. When the parameter is set to “true”, the device will make known to others that it has the Group Chat Full Store and Forward service capability using one of the methods in section 2.6.1.

2.6.3.5 File Transfer

In addition to the general user driven entry points and taking into account:

- The optimizations provided in section 2.6.1.5, and,
- The potential impact of having an SPI relationship between sender and receiver as described in section 2.6.2.4.

The capability exchange shall take place (when the service capability query is supported by SIP OPTIONS or the contact is not a VIP Contact for SPI):

- Before the initial SIP INVITE is sent to initiate the service to verify if the receiver is ready for the service (unless an exchange of capabilities has just been made based on one of the criteria listed in section 2.6.3.1, as described in section 2.6.4)
- If an error takes place:
  - After the service is cancelled either by the sender or receiver

- After the file transfer is abnormally interrupted as a result of a failure or irregular signalling behaviour during the establishment of the service is detected

2.6.3.6 Content sharing

In addition to the general user driven entry points and taking into account:

- The optimizations provided in section 2.6.1.5, and,
- The potential impact of having an SPI relationship between sender and receiver as described in section 2.6.2.4.

The capability exchange shall take place:

- Before the initial SIP INVITE is sent to initiate the service to verify if the receiver is ready for the service (unless an exchange of capabilities has just been made based on one of the criteria listed in section 2.6.3.1, as described in section 2.6.4)
- If an error takes place:
  - After the service is cancelled either by the sender or receiver
  - After the sharing is abnormally interrupted as a result of a failure or irregular signalling behaviour during the establishment of the service is detected
  - After the call is no longer active
- In any of the scenarios described in section 2.6.3.1 which are relevant to the service.

Additionally to the previous capabilities query entry point a client provider may implement additional ones to enhance the user experience. For example, it may be considered to issue a capability exchange when the relevant sensors in a device indicate that the user is likely to interact with the phone keyboard or screen during a call.

2.6.3.7 Social presence

Information indicating support for social information via presence is expected prior to a user's attempt to establish a social presence relationship. This supports the "Who Can I Invite" concept; providing the user with a view of the contacts with whom they can attempt to establish a social presence relationship. This information is provided in the following contexts:

- Discovery via SIP OPTIONS.
- Discovery via Presence

Independently of the chosen mechanism,

- If capability discovery indicates that both clients support the "social information via presence" functionality, the user is then presented with the possibility of inviting the contact to share the social presence information. This includes invitation of a previously discovered SPI-enabled contact who is temporarily Not Available. If not, the terminal should not present this possibility to the user for that contact.
- For those contacts who have an active social presence relationship established with the sender, it shall not perform a capability exchange if their capabilities are updated automatically using the standard SPI mechanisms described in section 3.7.4.

2.6.3.7.1 Discovery via SIP OPTIONS

To ensure interoperability[11] and enable those Service Providers implementing an SIP OPTIONS based capability/user discovery mechanism as default for their RCS deployments but deploying a Presence Server to additionally provide the social profile information (as described in section 3.7.4.2.2) functionality, the UE shall provide the following procedure:

> Prior to being able to send an invitation to a contact (e.g. from the address book), the terminal will use the OPTIONS mechanism to determine if the other end also supports this feature (that is both ends include the “Social Presence Information” SIP OPTIONS tag in the relevant headers).

2.6.3.7.2 Discovery via presence

Prior to being able to send an invitation to share Social Presence with a contact (e.g. from the address book), the terminal may use the Anonymous Fetch mechanism to determine if the other end also supports this feature (that is both ends include an “open” “Social Presence Information” Presence Service Tuple in the Presence Information Data Format [PIDF]). This includes inviting a contact who has previously been discovered to be Social presence-enabled even when they are currently offline.

2.6.3.8 IP Voice Call

2.6.3.8.1 IP Voice Call per MMTEL

The capability exchange is not required for this service. This capability may be used for network internal use and shall not have an impact on the user experience.

2.6.3.8.2 RCS IP Voice Call

In addition to the general user driven entry points and taking into account:

- The optimizations provided in section 2.6.1.5, and,
- The potential impact of having an SPI relationship between sender and receiver as described in section 2.6.2.4.

The capability exchange shall take place:

- Before the RCS IP Voice Call is initiated by the sender to verify if the receiver is ready for the service (unless an exchange of capabilities has just been made based on one of the criteria listed in section 2.6.3.1, as described in section 2.6.4)
- If an error takes place, after the call when the service was abnormally interrupted or irregular signalling behaviour during the establishment of the call is detected.
- In any of the scenarios described in section 2.6.3.1 which are relevant to the service.

2.6.3.9 IP Video Call

2.6.3.9.1 IP Video Call per MMTEL

In addition to the general user driven entry points and taking into account:

- The optimizations provided in section 2.6.1.5, and,
- The potential impact of having an SPI relationship between sender and receiver as described in section 2.6.2.4.

The capability exchange shall take place:

[11] Please note that the present specification allows the deployment of RCS communication services without the need for a Presence Server and the associated XDM servers, therefore, the present specification provide the necessary guidance to secure interoperability.

- Before the IP Video Call per MMTEL is initiated by the sender to verify if the receiver is ready for the service (unless an exchange of capabilities has just been made based on one of the criteria listed in section 2.6.3.1, as described in section 2.6.4)
- If an error takes place, after the call when the service was abnormally interrupted or irregular signalling behaviour during the establishment of the call is detected.
- In any of the scenarios described in section 2.6.3.1 which are relevant to the service.

2.6.3.9.2 RCS IP Video Call

In addition to the general user driven entry points and taking into account:

- The optimizations provided in section 2.6.1.5, and,
- The potential impact of having an SPI relationship between sender and receiver as described in section 2.6.2.4.

The capability exchange shall take place:

- Before the RCS IP Video Call is initiated by the sender to verify if the receiver is ready for the service (unless an exchange of capabilities has just been made based on one of the criteria listed in section 2.6.3.1, as described in section 2.6.4)
- If an error takes place, after the call when the service was abnormally interrupted or irregular signalling behaviour during the establishment of the call is detected.
- In any of the scenarios described in section 2.6.3.1 which are relevant to the service.

2.6.3.10 Geolocation PUSH

In addition to the general user driven entry points and taking into account:

- The optimizations provided in section 2.6.1.5, and,
- The potential impact of having an SPI relationship between sender and receiver as described in section 2.6.2.4.

The capability exchange shall take place:

- Before the initial SIP INVITE is sent to initiate the service to verify if the receiver is ready for the service (unless an exchange of capabilities has just been made based on one of the criteria listed in section 2.6.3.1, as described in section 2.6.4)
- If an error takes place:
    - After the service is cancelled either by the sender or receiver
    - If an error takes place and as a result the Geolocation PUSH is abnormally interrupted or irregular signalling behaviour during the establishment of the service is detected

2.6.3.11 Geolocation PULL

In addition to the general user driven entry points and taking into account:

- The optimizations provided in section 2.6.1.5, and,
- The potential impact of having an SPI relationship between sender and receiver as described in section 2.6.2.4.

The capability exchange shall take place:

- If an error takes place:
    - After the service is cancelled either by the sender or receiver
    - If an error takes place and as a result the Geolocation PULL is abnormally interrupted or irregular signalling behaviour during the establishment of the service is detected

2.6.4 Capability exchange optimisations

Depending on the circumstances and use cases, there could be occasions where the capability exchange may happen relatively often (in case of very frequent bearer changes for instance).

To avoid the overhead and increase the efficiency, the client may implement a mechanism to reduce the number of requests in situations where the capability exchange is likely to be performed too often. Examples of how this mechanism can be achieved are listed below:

- Introduce a degree of hysteresis (that is a capabilities update is sent/requested only when the circumstances which led to the change remain stable for a certain period of time).
- Implement a validity timer (that is if the latest capabilities we have were fetched less than X seconds ago, they are still considered as valid).

Please note that this specification does not describe detailed implementations to leave room for OEMs and third parties to drive innovative and differentiated solutions. This helps to distinguish their products from competitors.

2.6.4.1 Service Provider Controlled Service Capabilities Handling

The following items can be configured subject to the Service Provider's policies (see section A.1.10):

1. The maximum amount of capability query operations during a certain time period done by a client (that is, for all contacts)
2. An expiry of the capabilities for a specific contact
3. The Contacts considered for the capability discovery depending on their prefix (see CAPABILITY DISCOVERY ALLOWED PREFIXES defined in Table 91 in section A.1.10)

This will allow to control the maximum time before a client will discover that one of the contacts is now RCS capable

NOTE: there might be a conflict between the different provisioning settings controlling the frequency of capability query operations (for example, a too low maximum amount of fetch operation combined with a very low expiry time). In that case an RCS client will prioritize the maximum amount of fetch operations settings over the expiry. A Service Provider deploying RCS is likely to carefully consider the values of these settings and this is therefore not expected to be an issue in actual deployments.

2.7 Capability values and status

The RCS capabilities represent the list of services that an RCS user/client can access at a certain point in time. The capabilities depend on four factors:

1. User Service Provider provisioning status: A Service Provider may choose to limit service to customers depending on subscription status (e.g. chat and file share, but not video)
2. The terminal hardware (HW): A terminal with limited HW (i.e. no capability to process video) may not be able to access all the RCS Services
3. The terminal status: Even if a terminal HW supports all the services, it could be that the device status introduces a limitation (e.g. receiving files is not possible when the file storage is full)

4. Connectivity status: Some services may require a certain level of network Quality of Service (QoS). For example, streaming video over a 2G GPRS does not provide an adequate UX.

In addition to the factors presented above and as presented in Annex A section A.1, it is possible for a Service Provider to select which services are available for a particular user. Therefore, the previous considerations shall only be taken into account assuming that the relevant RCS services are enabled via configuration and consequently, Table 36 assumes that all the user's devices have been configured with all the RCS services enabled.

As a summary, please find the table below (note that it is assumed the client/terminal and the network supports each of the services as a precondition and that the client/terminal is provisioned to support all the services[12]):

Service	TERMINAL and STATUS REQUIREMENTS	Data Bearer					
		2G	EDGE	3G	HSPA	LTE	Wi-Fi
Standalone messaging	None	Y	Y	Y	Y	Y	Y
Chat (1-to-1 or group)	None	Y	Y	Y	Y	Y	Y
File Transfer (FT)[13, 14]	Minimum threshold of free space to store files	Y	Y	Y	Y	Y	Y
File Transfer via HTTP	The relevant configuration parameters are correctly set	Y	Y	Y	Y	Y	Y
Content share: Image Share	Minimum threshold of free space to store files. The terminal should be on an active call[15] with the user the image is willing to be shared with. Not available in multiparty calls.	Y[16]	Y[15]	Y	Y	Y	Y

[12] As presented in Annex A section A.1, it is possible for a Service Provider to select which services are available for a particular user.

[13] If a client supports to receive a thumbnail in the file transfer invitation as described in section 3.5.4 and FT THUMB (see section A.1.4) is set to enabled, it shall indicate the corresponding capability whenever the File Transfer capability is indicated. In that case both the File Transfer and the File Transfer Thumbnail capability shall be indicated.

[14] If a client supports file transfer store & forward as described in section 3.5.4 and FT STANDFW ENABLED (see section A.1.4) is set to enabled, it shall indicate both the File Transfer and File Transfer Store and Forward capability as described in section 2.6.1.3.1

[15] In this context, the term active call is used to indicate that a voice call is taking place with the user the image is shared with and that this call is not on-hold, waiting or forwarded/diverted. This limitation is not applicable for broadband access devices for the handling of a received capability request or an incoming invitation. The restrictions fully apply for outgoing requests.

[16] Note that it is only possible if device and the cellular network support Dual-Transfer Mode (DTM)

Service	TERMINAL and STATUS REQUIREMENTS	Data Bearer					
		2G	EDGE	3G	HSPA	LTE	Wi-Fi
Content share: Video Share during a call (IR.74)	Support video profile (encoding /decoding). The terminal should be on an active call[15] with the user the video is willing to be shared with. It is not available in multiparty calls.	N	N	Y One Way Only [17]	Y[18]	Y[18] Higher video profile	Y[18]
Content share: Video Share without a call (IR.84)	Support video profile (encoding /decoding).	N	N	Y[19]	Y[18, 19]	Y[18, 19] Higher video profile	Y[18, 19]
SPI	N/A	Y	Y	Y	Y	Y	Y
IP Voice Call [PRD-IR.92]/[PRD-IR.58]	N/A	N	N	N	Y (IR.58)	Y (IR.92)	N
IP Video Call [PRD-IR.94]	Support video profile (encoding /decoding).	N	N	N	Y (IR.94)	Y (IR.94)	N
RCS IP Voice Call	N/A	N	N	Y[20] RCS-AA or RCS-CS	Y[20] RCS-AA or RCS-CS	Y[20] RCS-AA or RCS-CS	Y[20] RCS-AA or RCS-CS
RCS IP Video Call	Support video profile (encoding /decoding).	N	N	Y[20] RCS-AA or RCS-CS	Y[20] RCS-AA or RCS-CS	Y[20] RCS-AA or RCS-CS	Y[20] RCS-AA or RCS-CS

[17] If on the current bearer sharing is supported one way only and a Video Share session is initiated by the device, a capability exchange should be performed to the other end to indicate that Video Share is no longer available. When the session is terminated or the bearer changes to one supporting bidirectional Video Share, the Video Share capability should again be announced.

[18] In this case both ends may share video simultaneously meaning that there is a possibility to have a bidirectional flow of video (see the other party's video while I am also sharing video with him/her). The meaning is that if a user is already sharing video with the other end, the other user may decide to also share video simultaneously, not that the two-ways Video Share can start simultaneously.

[19] Video Share without a call is always one way only (see section 3.6.1.3).

[20] Depending on Service Provider Policy

Service	TERMINAL and STATUS REQUIREMENTS	Data Bearer					
		2G	EDGE	3G	HSPA	LTE	Wi-Fi
Geolocation PUSH	Minimum threshold of free space to store files From the capability exchange point of view there are no additional terminal requirements however on the sender the service shall be only available if the terminal (UE) provides a mean to access the location information required for the service.	Y	Y	Y	Y	Y	Y
Geolocation PULL	Primary device with capability for locating	Y	Y	Y	Y	Y	Y

Table 36: RCS services: Terminal, status and data bearer requirements

2.7.1 Additional considerations for specific RCS services

2.7.1.1 Chat store and forward: Impact in the capability exchange

As presented in section A.1.3.3 (IM CAP ALWAYS ON), there is the possibility to configure the client to assume that the Service Provider will be providing the Chat store and forward functionality, which consists of storing messages which are sent to users who are offline (i.e. no data connectivity or device off) at the time the chat message is sent.

If this parameter is enabled, there is an impact from the Chat capability which is presented to the user.

As a consequence, we have 4 different types of contacts for Chat capability:

ID	Targeted contact is RCS Chat capable?	Originating Service Provider supports Store& Forward?	Targeted contact is connected to the network?	Impact on starting Chat
1	NO	N/A	N/A	Chat with that contact is only possible if interworking is provided (see IM CAP NON RCS in Table 85)
2	YES	NO	NO	Not possible to start a Chat at that time
3	YES	YES	NO	Possible to send a Chat message that will be delivered later by the Store and Forward server as soon as the Contact is connected
4	YES	Not Relevant	YES	Chat is possible and messages are immediately delivered

Table 37: Store and forward possible scenarios

The Chat behaviour on the client is controlled by these configuration parameters (see Annex A for further information):

- IM CAP ALWAYS ON:
  When a Service Provider implements store and forward, they may choose to provision all the RCS users with the IM CAP ALWAYS ON configuration parameter set to enabled. This means that all RCS contacts (currently registered or not) are presented with the Chat service as available (3 and 4 according to Table 37). This may also be the case when the device is offline itself. In that case the composed messages should be queued and sent as soon as the device registers with the service again. When the messages are included in the SIP INVITE request (see section 3.3.4) a provisional response (including 100 Trying) should be received on the SIP INVITE request before sending a subsequent queued message. If a session is established (i.e. a 200 OK response is received on the SIP INVITE request that may or may not have included a message), the remaining queued messages targeted at that contact shall be sent in the established MSRP session where a following queued message should be sent as soon as a MSRP 200 OK has been received on the last one.
  When store and forward is not implemented by the SP, all its RCS customers will have the IM CAP ALWAYS ON configuration parameter is set to disabled (2 and 4 according to Table 37).
  As a summary: IM CAP ALWAYS ON is enabled when store and forward functionality is provided in the network, otherwise it is disabled

- When IM CAP ALWAYS ON is enabled, IM WARN SF can be used to control the UI behaviour:
  If IM WARN SF parameter is enabled: In scenarios 3 and 4, the user shall be made aware that messages delivered to unregistered users will be only delivered once the other party is back online (for example after switching on the device or regaining network coverage). A user shall be considered as unregistered when the received presence document did not include any services in case a presence based capability check was performed or the OPTIONS request resulted in an inconclusive response (i.e. 480 or 408) or a response that included the *automata* tag defined in [RFC3840].
  If IM WARN SF parameter is disabled, there shall not be any visible difference between scenarios 3 and 4 from the UI point of view. Therefore, the user shall not be made aware of whether the messages are being stored or are delivered directly to the other party.
  When IM WARN SF is enabled, the device/client uses the response from the capability exchange to determine whether a warning is displayed to the user.

When interworking of chat to SMS/MMS is available for users, two more parameters can be used in a similar way: IM CAP NON RCS and IM WARN IW. More information can be found in section A.1.3.3.

2.7.1.2 Video and Image Share additional considerations

2.7.1.2.1 Bidirectional Video Share

Bidirectional Video Share means that once User A is sharing a video with User B and providing the right coverage conditions are in place, User B could also start to share a video with User A simultaneously. In this case each Video Share session is independent and is handled separately. When a device moves from a bearer that supports this bidirectional Video Share to a bearer that only supports one way sharing (e.g. from HSPA to 3G) and there is an active Video Share session in each direction, the device that changed bearers shall terminate the Video Share session that it initiated itself.

For clarification purposes, the following assumptions are made for the Image and Video Share cases:

- Both the sharing and receiving end are in a call (that may for instance be CS) between them
- The call is not a multiparty call
- The call is not on hold
- The call is not waiting
- A call forward or divert is not in place

Meaning the relevant Image and Video Share tags described in section 2.6.1.1.2 shall be included only if:

1. The OPTIONS exchange happens when the user is on an active call, and,
2. The destination (sending OPTIONS) or the requester (receiving an OPTIONS message which has to be replied with a response) is on the other end of the active call, and,
3. Network coverage supports sharing (see section 2.7), and,
4. Either bidirectional sharing is supported or the device has not initiated a sharing session itself.

Also for clarification, provided other RCS services (e.g., Standalone Messaging, Chat, File Transfer) are available (e.g. the conditions of coverage and space are met and the device UI supports these services simultaneously with the call), the relevant service capability tags should be included with the Image and Video Share tags.

Note that while capability exchange is reciprocal, User A and User B's capabilities may be different and services shall be made available accordingly (e.g. User A may support video encode and User B may support decode, but both need to be under 3G or better data coverage for the service to operate).

In addition to the information presented above, it should also be taken into account that some terminals do not support 2G DTM (Dual-Transfer Mode). When such devices are within a 2G data coverage (meaning that no services are available during the call), the PS connection will automatically drop once they engage in a CS call.

NOTE: Information on codec support for Video Share is covered in section 3.6.

2.7.1.2.2 Image orientation extension for video

When capturing video on mobile devices, frequently the orientation of captured frames will not be "facing up" relative to the preview display on-screen. For example, many devices have their cameras oriented in landscape mode, while for video conversation scenarios it is more appealing to show a portrait user interface. This means that the image captured by the camera will be rotated 90 degrees clockwise (CW) or counter-clockwise (CCW). The remote end, however, cannot know a priori how to rotate the image and thus we must define a method to transmit this auxiliary information.

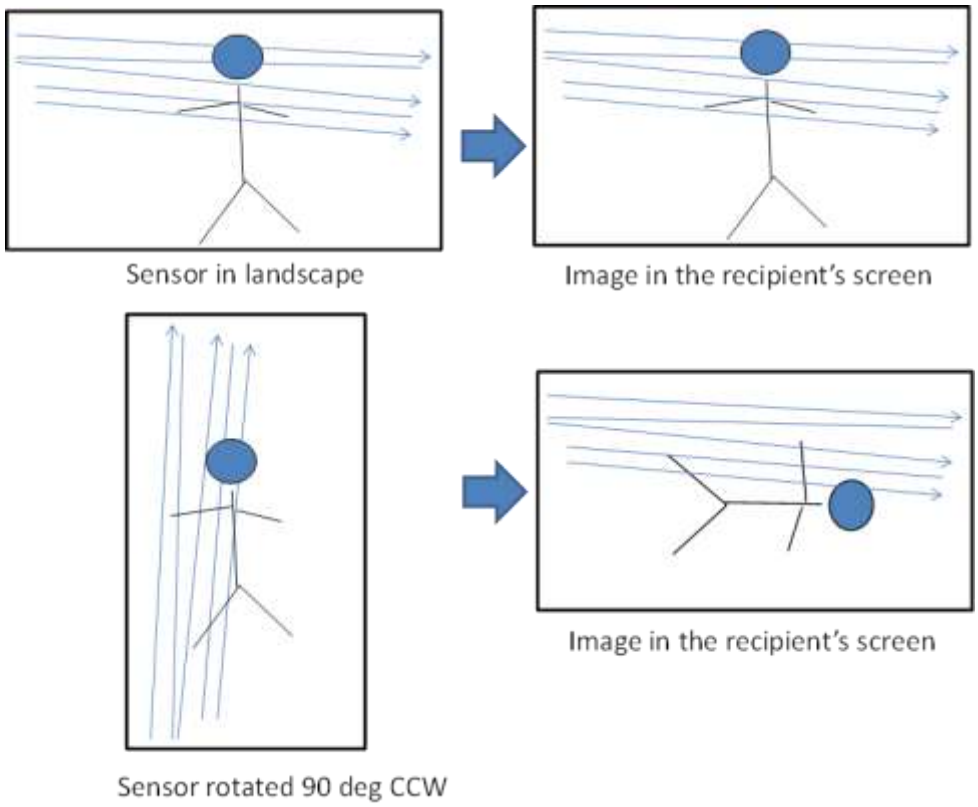


Figure 32: Illustration of the video-orientation problem

To prevent that from happening, an RCS device shall implement the following Coordination of Video Orientation (CVO) mechanism described here but also defined in [3GPP TS 26.114]:

NOTE: The mechanism described here is exactly what is also described in [3GPP TS 26.114].

- Relying on the mechanism defined by [RFC3550], [RFC5285] and [3GPP TS 26.114] by which generic information can be passed in-band with the RTP stream provided that

both sides agree out-of-band on the 4-bit ID (IDentifier) that will be used to designate it, following URI will be used to indicate support of this CVO mechanism: *urn:3gpp:video-orientation*.
An example of what SDP with support for this extension looks like is provided below:

```
v=0
o=- 1323909835 1323909838 IN IP4 10.0.100.189
s=-
c=IN IP4 10.0.100.189
t=0 0
m=video 4284 RTP/AVP 118
a=sendonly
a=rtpmap:118 H264/90000
a=fmtp:118 packetization-mode=0;profile-level-id=42900b
a=extmap:7 urn:3gpp:video-orientation
```

Table 38: Image orientation for video support: SDP sample

The number 7 is arbitrarily chosen by the sender, as defined in [RFC5285]. When answering the INVITE, the remote end will add the same line a=extmap:7 urn:3gpp:video-orientation to its SDP, to signal that it supports this mechanism as well.

To ensure maximum backward compatibility with clients not supporting the image orientation extension the RCS client shall exclude the *extmap* attribute in the SDP answer if the SDP offer does not contain the *extmap* attribute. Conversely, the RCS client SHALL send the extra RTP header only if the received SDP from the remote peer contains the *extmap* attribute.

- The additional RTP extension header defined here is to be added to the last RTP packet in each group of packets which encode a key frame (I-frame or IDR frame in H.264), to avoid adding too much overhead to the stream. The additional RTP extension header may also be added onto the last RTP packet in each group of packets which make up another type of frame (e.g. a P-Frame) only if the current value is different from the previous value sent. If this is the only header extension present, a total of 8 bytes are appended to each key frame, and the last packet in the sequence of RTP packets will be marked with both the marker bit and the Extension bit, as defined in [RFC3550].
  - Packet loss may cause the partial loss of a key frame, and thus the loss of the information in the extension header. However, since the video decoder is unable to reliably display any frames once it detects that a key frame was lost, the loss of the device orientation information is not significant. It is assumed that the same recovery mechanism used to restore the video stream will be used to retransmit the device orientation header.
- The proposed extension header data has a total length of one byte:

7	6	5	4	3	2	1	0
unused				CAM	ROT		

Table 39: Usage of the proposed RTP extension header usage

  - The first four Most Significant Bits (MSBs) in the extension header data are not used and remain reserved for future extensions.
  - The fourth bit (CAM) signals which camera is currently in use:
    - 0: Front-facing camera
    - 1: Back-facing camera

- The remaining three bits (ROT) are used to indicate the image orientation within the video as it is transmitted on the wire. The RCS client obtains the ROT value based on a combination of:
  - The relevant sensors that indicate the rotation of the device, together with,
  - The default camera orientation.

  Details are provided in Table 40 and Table 41:

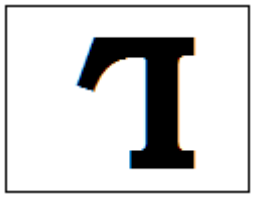
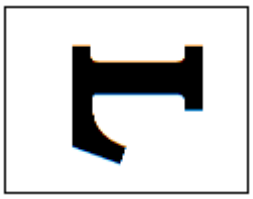
ROT	Image orientation as sent on the wire	Actions on the receiver before display
000		None
001		Rotate 90 degrees CW
010		Rotate 180 degrees CW or CCW
011		Rotate 90 degrees CCW

Table 40: Usage of the orientation bits (ROT) in the RTP extension (1/2)

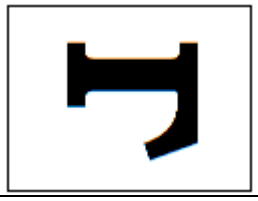
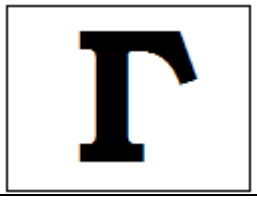
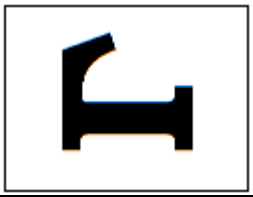
ROT	Image orientation as sent on the wire	Actions on the receiver before display
100		Flip horizontal
101		Rotate 90 degrees CW Flip horizontal
110		Rotate 180 degrees CW or CCW Flip horizontal
111		Rotate 90 degrees CCW Flip horizontal

Table 41: Usage of the orientation bits (ROT) in the RTP extension (2/2)

- Thus the individual bits can be assigned the following meaning:
  - Bit 0 (Least Significant Bit, LSB): Rotate 90 degrees CW
  - Bit 1: Rotate 180 degrees
  - Bit 2: Flip horizontally.
- The order of operations is important, and must be performed in the listed order (i.e. LSB to MSB).

2.8 RCS protocols

The following table summarises the list of protocols employed by RCS clients. It must be noted that the choice among the options presented will not impact Service Provider interoperability:

Protocol name	Description	Transport layer	Secure transport layer/protocol
Session initiation protocol (SIP)	Client-IMS core signalling protocol	User Datagram Protocol (UDP) over IP or Transmission Control Protocol (TCP) over IP	SIP over Transport Layer Security (TLS) or IP Security (IPsec)
Message Session Relay Protocol (MSRP)	chat messages, media (pictures) and file exchange protocol	TCP/IP	MSRP over TLS
Real-time protocol (RTP)	Real Time Media (voice and video) exchange	UDP/IP	Secure RTP (SRTP) (see [RFC3711])
Internet Mail Access Protocol (IMAP)	Access to Message Store Server	TCP/IP	IMAP over TLS
Hyper Text Transfer Protocol (HTTP)	XML configuration access protocol (XCAP) transactions HTTP configuration mechanism	TCP/IP	HTTPS
Secure User Plane Location (SUPL) Transport Protocol	Geolocation positioning	UDP	TLS

Table 42: RCS protocols

According to [RFC3261] RCS clients shall support both SIP/UDP (User Datagram Protocol) and SIP/TCP (Transmission Control Protocol). The choice of whether both are used or only TCP is used to transport the signalling data belongs to each Service Provider and is controlled by the configuration parameters "*psSignalling*" and "*wifiSignalling*" in Annex A section A.2.10.

Regarding the impact of Network Address Translation (NAT) traversal in the different protocols involved in RCS, the following considerations shall be taken into account:

- Regarding the SIP protocol:
  - Carriage Return Line Feed (CRLF) keep-alive [RFC6223] support is MANDATORY when only SIP/TCP or SIP/TLS is used by the RCS client and SIP/UDP is not used. Section C.1 describes how both client and server could initiate the sending of the keep alives.
  - Simple Traversal of UDP through NATs (STUN) keep-alive [RFC6223] support is RECOMMENDED when SIP/UDP is used by the RCS client as it allows network capacity optimization.
  - An RCS client using SIP/UDP:
    - Shall support symmetric signalling (That is the IP and port combination used to send SIP messages is the same as the one used to receive SIP messages).
    - Shall perform TCP switchover for large SIP messages.
- For handling Message Session Relay Protocol (MSRP) sessions, the RCS MSRP endpoints shall support:

  - [RFC6135]: “The Alternative Connection Model for the Message Session Relay Protocol (MSRP)”
  - The mechanisms described in section 2.8.2 regarding session matching for MSRP.
  - For NAT traversal for MSRP, keep alives (i.e. empty MSRP packets) are not necessary. If the TCP connection is torn down because of inactivity, the MSRP session is torn down, and a new SIP INVITE request to set up a new MSRP session is sent the next time a message is to be sent.
- Regarding NAT traversal of Real-Time Transport Protocol (RTP) sessions, the RCS client should implement the mechanism described in section 2.8.1.
- For Internet Mail Access Protocol (IMAP), HTTP and Secure User Plane Location (SUPL) no specific mechanisms are mandated in the current specification to support NAT traversal

The support of Transport Layer Security (TLS) based or IP Security (IPsec) based protocols to secure the signalling and TLS based for MSRP and IMAP protocols or Secure Real-Time Transport Protocol (SRTP) for RTP protocols to secure media exchanges is RECOMMENDED particularly for those scenarios where the data is carried over a network outside the Service Provider domain (i.e. Wi-Fi access). For more information on access security, see section 2.13.

NOTE: To ensure interoperability of all RCS capable devices across different Service Provider networks, the list of preferred options for the transport and security for the signalling and media protocols is included in the configuration parameters as defined in Annex A, section A.2.10. Consequently, a Service Provider provides this information as part of the first-time or re-configuration scenarios described in section 2.3.

2.8.1 RTP and NAT traversal

As mentioned previously, an RCS client must implement several mechanisms to avoid the negative impact of NAT traversal, which can both occur when connecting over:

- PS: Mainly due to the scarcity of IPv4 public addresses and proxying performed at APN level, or,
- Wi-Fi: In this case due to the fact the network topology between the access point and the Internet may vary between deployments.

To combat the negative effects of NAT traversal on the RTP protocol, the RCS client:

- Shall support a keep-alive mechanism to open and maintain the NAT binding alive regardless of whether the media stream is currently inactive, send-only, receive-only or send-receive. The recommended standard keep-alive mechanism is an empty (no payload) RTP packet with a payload type of 20 (as per [3GPP TS 24.229]).
  - SHALL when sending empty packets instead of using STUN and it is about to receive a Video Stream send these dummy RTP packets at a high rate (recommended rate: 50 to 100ms) from the moment the SIP INVITE request is received (or the 180 RINGING is sent) in bursts sent regularly (a 1 second burst every 15 seconds is recommended). This shall be done until one of the following conditions is met:
    - The first RTP packet of a Video Stream is received or
    - The client starts streaming itself in case of a bi-directional RTP stream or
    - A final response is sent on the SIP INVITE request. In case this final response is a 200 OK response, the client shall continuously send the dummy RTP packets

until either the first RTP packet of a Video Stream is received or the client starts streaming itself in case of a bi-directional RTP stream.

Once the first RTP packet is received the dummy packets shall be sent at a lower rate (a transmission every 15 sec is recommended) for the remainder of a uni-directional session or not at all in case the RTP stream is bi-directional.

- If the first frame is not an I-Frame or Network Abstraction Layer (NAL) unit carrying a Sequence Parameter Set (SPS) or Picture Parameter Set (PPS), the receiving client SHALL send a RTCP Full Intra Request (FIR) (see [RFC5104], section 4.3.1) to the sender
- SHALL reset the encoder as specified in [RFC5104] when receiving an RTCP FIR, and send SPS, PPS (if not provided in the SDP) and an I-Frame to the receiver

- shall use symmetric media (that is use the same port number for sending and receiving packets) as defined in [RFC4961] mechanism which is summarized below:
  - When an invitation for Video Share is received and accepted, the 200 OK response contains a SDP body containing all the necessary fields (including the destination port) for the sender to send the RTP packets.
  - Immediately after sending the 180 Ringing response, the receiver will send a keep-alive packet back to the sender to secure the timely setup of the media path:
  - The source port shall be identical to the one included in the m field of the SDP payload inside the 200 OK response.
  - The destination port shall be identical to the one included in the m field of the SDP payload inside the SIP INVITE message.
    - The sender should allow enough time for the media path to be secured.

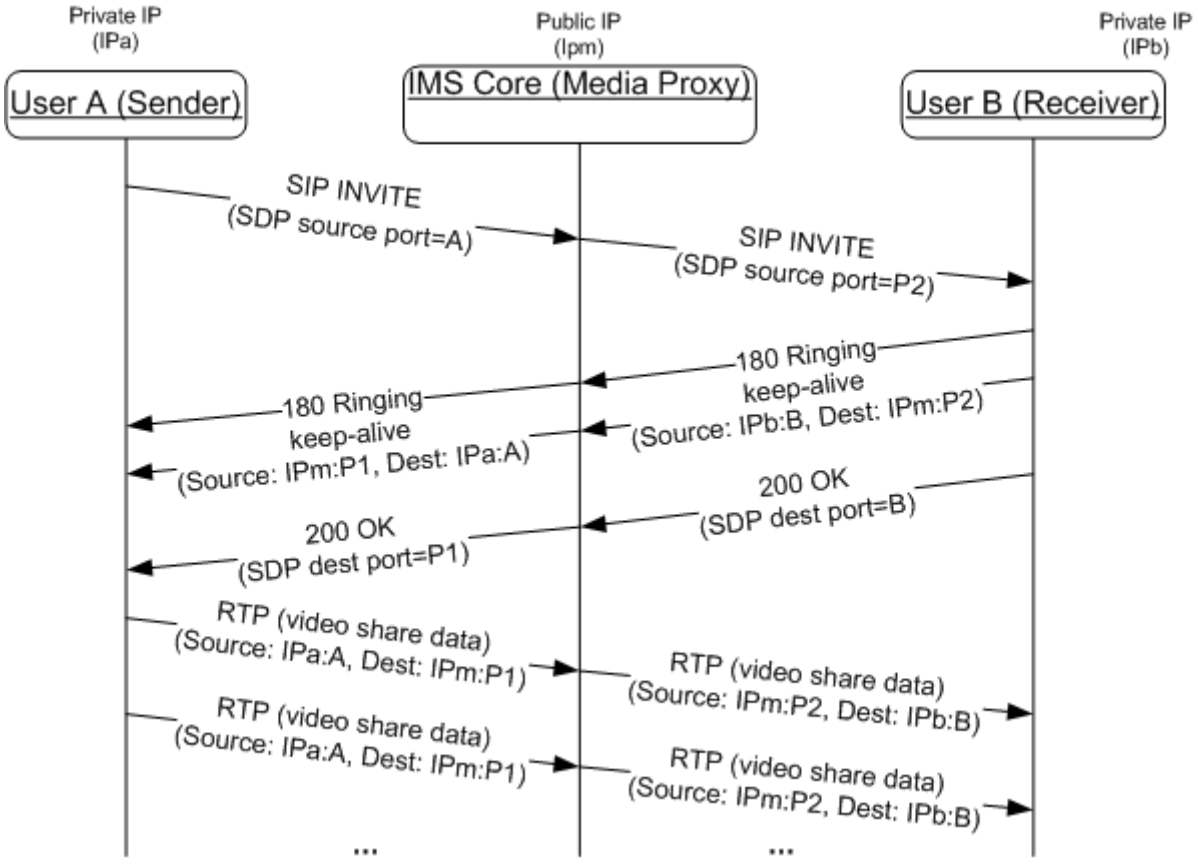


Figure 33: RTP symmetric media path establishment

NOTE1: as a general recommendation, User A should also send a keep-alive once it receives the SDP from the other side.

- shall use the Real-Time Transport Control Protocol (RTCP)

  The symmetric media procedure described for the RTP protocol is, in general, applicable to any UDP stream. As the usage of RTCP is also mandatory, an analogous mechanism shall be implemented to prevent any RTCP streams from being blocked. Therefore, the symmetric media procedure described in this section for RTP is also applicable to RTCP and shall be employed (that is a dummy packet is sent by the receiver to secure the RTP flow and a second one is used to secure the RTCP flow). Also the sender device/client shall send a dummy packet when the session is established to secure the RTCP flow on their side and ensure the reception of any RTCP RR (Receiver Report) sent by the receiving side. The dummy packet format recommended for establishing the RTCP flow is an empty RTCP RR or empty RTCP SR (Sender Report).

  NOTE2: For a VoLTE/VoHSPA enabled device, RTCP usage for a voice session shall be as defined in section 3.2.4 of [PRD-IR.92]

Please note that for readability purposes, the procedures described in this section have not been included in the diagrams in section 3.6 covering the Video Share functionality.

2.8.2 MSRP session matching

NOTE: The text in this section is based on the text contained in the now expired IETF internet draft draft-ietf-simple-msrp-sessmatch-10.

When the CHAT MESSAGING TECHNOLOGY configuration parameter defined in Table 85 is set to use OMA CPM, an RCS client shall set up MSRP sessions as per [RCS5-CPM-CONVFUNC-ENDORS]. Otherwise, the session shall be set up according to the procedures in [RFC4975], [RFC6135] and the procedure in this section for the session matching.

This section defines how an MSRP entity (e.g. an RCS Client, Messaging Server or other node handling MSRP within the network) that does not support the procedures in [RCS5-CPM-CONVFUNC-ENDORS] or is configured not to use them matches an incoming MSRP message to an MSRP session. The difference between the session matching mechanism in [RFC4975] and the one defined here is that while the mechanism in [RFC4975] uses the MSRP URI comparison rules for session matching, for RCS, only the session-id part of the MSRP URI is used.

When an MSRP entity receives the first MSRP request for an MSRP session, the To-Path header field of the request should contain a URI with a session-id part that was provided in the SDP associated with the MSRP session. The entity that accepted the connection looks up the session-id part of the MSRP URI in the received requests, in order to determine which session it matches. The session-id part is compared as case sensitive. If a match exists, the entity shall assume that the host that formed the connection is the host to which this URI was given. If no match exists, the entity shall reject the request with a 481 error response. The entity shall also check to make sure the session is not already in use on another connection. If the session is already in use, it shall reject the request with a 506 error response.

2.8.3 SIP Issues

1. An RCS client should use a random originating SIP signalling port of the range 1025-65535. If the selected port is not available, the next port number shall be used for this session.

2. An RCS client shall build its SIP contact address to be unique. A recommended way to do so is to use a hashed value of the +sip.instance tag as user part of the URI of the contact address.
3. For an incoming request, an RCS client should verify that the Request-URI matches the URI of its registered contact address. If not, the Request-URI shall be considered an unexpected address and the request shall be rejected as per [RFC3261] section 8.2.2.1.

2.9 RCS and Access Technologies

2.9.1 RCS and Cellular Access

A device capable of supporting RCS-VoLTE or RCS-VoHSPA mode (see section 2.2.1) shall implement the domain selection function as described in [PRD-IR.92] and [PRD-IR.58]. The domain selection function selects whether the CS or IMS domain is used for the voice service. When CS domain is selected, the device is in RCS-CS mode, and when IMS domain is selected, the device is in RCS-VoLTE mode (when on LTE) or RCS-VoHSPA mode (when on HSPA).

The home Service Provider can use the configuration mechanisms defined in section 2.3.3 to configure the IMS Management Object defined in [3GPP TS 24.167] (see section A.1.6) to set the parameter “Voice_Domain_Preference_E_UTRAN” to value 1 “CS Voice only”. When this parameter is set to “CS Voice only”, the device shall revert to provide only RCS-CS mode (see section 2.2.1). A device configured as "CS voice only" shall behave as a device in RCS-CS mode with regards to the registration in the IMS. Only a Service Provider who supports both VoLTE and RCS needs to provide this setting. The device will determine the domain used for voice as specified in [3GPP TS 23.221].

Note that when "CS Voice only" is used, and if CS Fallback (SGs interface) is not supported in the LTE network the voice-centric UE will not use LTE.

The aim of the present section is to give an overview of the possibilities to complement and integrate LTE, High Speed Packet Access (HSPA) and RCS.

2.9.1.1 Access used by RCS in relation to VoLTE/VoHSPA

For a device capable of VoLTE/VoHSPA which is configured to use VoLTE/VoHSPA (see section 2.2.1) (i.e., that allows RCS-VoLTE or RCS-VoHSPA mode), LTE/HSPA is used for RCS features. VoLTE/VoHSPA is assumed to be natively implemented and integrated within the device and used for telephony when camping on LTE/HSPA. The IMS registration shall be shared between VoLTE/VoHSPA and RCS if the device is under VoLTE enabled LTE coverage or VoHSPA enabled HSPA coverage. A device which has performed SR-VCC is considered to be a device in RCS-CS mode and continues to keep the same IMS registration.

For a device in RCS-CS mode that is not configured to use VoLTE, CS is used for telephony and LTE access is used for RCS features provided that the device is camping in those networks and capable of using those PS radio access technologies.

A device in RCS-CS mode with LTE access will fall back to CS for telephony calls. Once CS fallback occurs, LTE access is dropped, and RCS functionality is provided via 3G/2G access until the call finishes; at which point the device may return to LTE access again if current coverage conditions allow.

A device in RCS-CS mode with HSPA access will use CS for telephony calls, and RCS will continue to use HSPA.

A device configured to use VoLTE/VoHSPA that is not on a network supporting VoLTE/VoHSPA, is a device in RCS-CS mode; LTE or HSPA access is used for RCS

features provided that the device is camping in those networks and it will fall back to CS for voice calls.

Once CS fallback occurs, LTE access is dropped and RCS functionality is provided via 3G/2G access until the call finishes; at which point the device may return to LTE access again if current coverage conditions allow. HSPA can continue to be used. These devices always use the IMS APN for accessing the RCS services.

2.9.1.2 LTE and HSPA Radio Capabilities

Radio bearers, UE Discontinuous Reception (DRX) and Discontinuous Transmission (DTX) modes of operation, Radio Link Control (RLC) configurations, and Guaranteed Bitrate (GBR) and Non-Guaranteed Bitrate (NBGR) services, GBR Monitoring Function and Conversational Traffic Class Handling are all as specified in [PRD-IR.92] and [PRD-IR.58] for devices in RCS-LTE and RCS-HSPA mode respectively. None of this is applicable to the devices in RCS-AA or the RCS-CS mode.

2.9.1.3 Bearer aspects

For all IMS traffic the following applies for an RCS device configured for VoLTE/VoHSPA:

- APN usage shall be according to [PRD-IR.92] or [PRD-IR.58] section 4.3.1 including the one used for XCAP

  NOTE1: For the APN to be used for XCAP, IMAP and other supporting protocols in the context of RCS see section 2.9.1.4.

- For LTE and HSPA bearer management see section 4.3 of [PRD-IR.92] and [PRD-IR.58] respectively.

For all RCS IMS traffic the following applies:

- For a device in RCS-VoLTE mode: LTE QCI (QoS class identifier) 8 and 9 shall be supported so that either may be used for MSRP traffic.
- For a device in RCS-VoHSPA mode following bearers shall be supported so that either may be used for MSRP traffic:
  - Universal Mobile Telecommunications System (UMTS) bearer with interactive traffic class, Traffic Handling Priority (THP) 3 and no Signalling, and
  - UMTS bearer with background traffic class

  NOTE2: For HTTP, XCAP and IMAP similar QoS categories should be used. Feedback has been requested from GSMA IREG on the way in which to establish the bearers to realize this.

For a device in RCS-AA mode or a device that is not configured for VoLTE/VoHSPA which is in RCS-CS mode the following applies for all RCS IMS traffic:

- When connecting through cellular access the device in RCS-AA or RCS-CS mode shall use an APN obtained through client configuration

  NOTE3: the Service Provider should ensure that the configured APN can handle all protocols used for RCS, including XCAP, HTTP and IMAP;

- For other ways of connecting by devices in RCS-AA or a device that is not configured for VoLTE/VoHSPA which is in RCS-CS mode (e.g. Wi-Fi, fixed broadband and so on): no requirements.

2.9.1.4 APN and roaming considerations

General technical guidelines on how roaming is handled for the RCS services shall follow [PRD-IR.65].

Guidance given for RCS and access technologies as documented in Chapter 2.9 are applicable also in the roaming scenarios. Specific roaming considerations for the different RCS device types (as specified in section 2.2.1):

- All services on a device configured for VoLTE, whether it is in RCS-VoLTE mode or in RCS-CS mode, shall follow the general rules as per [PRD-IR.88], APN usage as per [PRD-IR.92]
- All services on a device configured for VoHSPA, whether it is in RCS-VoHSPA mode or in RCS-CS mode, shall follow the general rules as per [PRD-IR.33], APN usage as per [PRD-IR.58]
- A device in RCS-AA mode or RCS-CS mode: no specific requirements

The APN to be used to access the RCS services[21] depends on the capacity of the device and the network to support an IMS APN as per [PRD-IR.88] and on the device configuration:

- When the device and the home network support the use of the IMS APN, the IMS APN shall be used to access the RCS services when the device is configured to support RCS-VoLTE or RCS-VoHSPA mode when possible or is configured through the ALWAYS USE IMS APN configuration parameter defined in section A.1.10 to use the IMS APN;
  When roaming on a network where the device cannot access a local IMS APN (e.g. no VoLTE roaming agreement is in place), a client configured to use the IMS APN will, by using the IMS APN, automatically access RCS through the home network's IMS APN with the telephony service using the Circuit Switched network.
  To support traffic from non-RCS applications (e.g. generic internet access) in this case the device and network shall support other APNs to be active simultaneously.

  NOTE1: The APN to use for HTTP, XCAP and IMAP when using the IMS APN has not been defined. The IMS APN itself cannot currently be used for those protocols because solutions are missing to handle them during roaming and to set the required QoS level (see section 2.9.1.3). Feedback has been requested from IREG on how to resolve this situation and when received, this specification will be updated to provide an approach for handling those protocols.

  NOTE2: Similarly it is not clear what APN to use for the configuration request on a non-configured device. Until a solution has been included detailing the handling of HTTP in combination with the IMS APN, the device's generic data access APN (i.e. the internet APN[22]) shall be used. It is out of scope of this specification how that APN is configured on the device.

- For other cases, either the Internet APN or an APN to be used only for RCS shall be used when accessing via PS (i.e. not accessing via Wi-Fi)
  - For these devices and within the scope of the RCS services, a new APN is defined: the RCS only APN;
  - Analogously to the IMS APN defined in [PRD-IR.88] the RCS only APN only provides access to the IMS services, and in this particular case, only to RCS services;

[21] This section only covers the APN behaviour for RCS services. These settings shall not be taken into account for the usage of other APNs by non-RCS services.

NOTE3: As the RCS only APN will only allow RCS related traffic, it will discard other requests. The device can therefore send all traffic on this APN and rely on the APN's filtering.

- The RCS only APN is configured via the RCS-E ONLY APN parameter presented in section A.1.11 in Table 92;
- If the RCS-E ONLY APN parameter has no value, then only the Internet APN[22] shall be allowed in the home network and a roaming network.

A switch at UI level to enable/disable PS connections for RCS may be shown upon a Service Provider decision (as per the value of the "ENABLE RCS-E SWITCH" RCS client configuration parameter defined in Annex A, section A.1.11). The purpose of this local client switch is to protect users from unexpected charges, especially when roaming.

The behaviour of the RCS Switch is different depending on whether the device and network support the IMS APN:

- If the IMS APN is supported, the behaviour is shown in the following table:

RCS Switch	APN to use for RCS services	Result
Disabled	IMS APN	Among the RCS services, the client shall only register in IMS for non-RCS IP Voice Call (only [PRD-IR.58]/[PRD-IR.92]), non-RCS IP Video Call (i.e. [PRD-IR.94]) and Standalone Messaging (see section 3.2).
Enabled or not available	IMS APN	Standard configuration, the client shall register in IMS for any supported RCS services

Table 43: APN configuration proposal for RCS for a device supporting the IMS APN

- If the IMS APN is not supported, the RCS Switch will determine whether the client shall register or not for RCS services as shown in Table 44:

In addition to this and in devices that do not support simultaneous use of the Internet and RCS only APN, two different APNs are considered:

1. The Internet APN, and,
2. The RCS only APN that, similar to the IMS APN defined in [PRD-IR.88], only provides access to the IMS services, and in this particular case, to RCS services.

In such devices the user shall be able to configure to allow or disallow RCS and/or internet traffic in the device settings when roaming according to the following alternatives:

Data traffic switch (combination of main data switch and roaming data switch)	RCS switch	APN to use for RCS services	Result
Enabled	Disabled	N/A	RCS client shall not register on the IMS network.

[22] By Internet APN, we understand the default APN configured by the Service Provider to provide Internet connectivity on the device.

Enabled	Enabled or not available	Internet APN	Standard configuration
Disabled	Enabled	RCS only APN	RCS only configuration This configuration is only available if the RCS-E ONLY APN is configured to a non-empty value
Disabled	Disabled	None	No data configuration

Table 44: APN configuration proposal for data traffic and roaming

2.9.1.4.1 Data connection notifications

For a device enabled for VoLTE, the device will be responsible for initiating a PS connection using the required APN and it should not be necessary to notify the user.

In other cases, taking into account the regulatory frameworks applying to some markets, it could be necessary to notify the user when a PS connection is going to be initiated. From the data connection notification point of view, there are three possible configurations:

Setting	Terminal behaviour
never connect	• connection disabled • no pop-up
always ask	• pop-up*: requesting confirmation to go online and informing about possible data charges • user has the following options: reject, confirm to connect once or to switch to 'always connect' and connect • when user confirms the connection is enabled *Alternatively, a shortcut to the device data settings, together with a warning that data charges might apply, is presented where the user may enable the connection.
always connect	• connection enabled • no pop-up

Table 45: Data connection notification options

Consistently with the configuration switches presented in the previous section (RCS on/off, data on/off), an RCS device shall be able to apply the data connection notification options (described in Table 45) individually to each of the following connections:

- Internet home: Standard data connection occurring within the Service Provider's home network.
- Internet roaming: Standard data connection when roaming.
- RCS home: Data connection required for RCS occurring within the Service Provider's home network.
- RCS roaming: Data connection required for RCS when roaming

Regarding the data connection switches presented in section 2.9.1.4, it is to the decision of each Service Provider to define during customization on whether to:

- Define the default settings ("always connect" for the "home" connections and "always ask" for the "roaming" connections)
- Define if the data connection notification settings are shown as part of the device configuration settings (that is the user is able to change the notification behaviour) instead.

2.9.2 Other access networks

2.9.2.1 Overview

In addition to the cellular PS access networks described in sections 2.9.1.4 and 2.9.1, the RCS framework and services can be used over any IP access over which the Service Provider's IMS core and application servers can be reached, provided that it offers sufficient bandwidth and an acceptable latency. Section 2.7 provides a guideline for which services can be used when connected through different types of access networks including broadband access.

These other networks can be both trusted and untrusted networks. "Trusted" means that HPLMN considers the access network trusted independently of the specific mechanism for authentication and encryption of end-user traffic that the access network implements. The access network is integrated into the Service Provider core infrastructure in such a way that the whole path from a mobile to services is considered secure by the HPLMN and under the control of respective Service Providers. Fixed access networks including ADSL (Asymmetric Digital Subscriber Line), cable modem access, FTTH (Fibre To The Home) and WLAN networks could therefore be considered as a "Trusted" network by the HPLMN if they are entirely Service Provider controlled. The same holds for clients using a cellular PS connection as broadband access.

Untrusted broadband networks however are at least partly controlled by some 3rd party and may therefore require more elaborate security measures to guarantee privacy and authenticity of the signalling and media traffic. As in this case the network does not provide the support for functionality such as encryption natively, it needs to be added to it before that particular network can be used to access the IMS core system. If the HPLMN considers direct access from a broadband network such as public Wi-Fi hotspots to the IMS core system untrusted, then in order to avoid security risks such as Denial of Service attacks towards Service Provider core components there is a requirement for additional secure access mechanisms to be deployed.

Many such secure access mechanisms are possible and can either be xSIM based or use other types of credentials. For commercial deployments the choice will be dependent on the type of client and on the environment. For example the same type of client could only use PS Mobile Broadband Access, access over the internet or only over a fixed ADSL line / FTTH access which may require using different mechanisms for each case. Therefore given that this choice will end up being specific to each deployment, it is not considered in scope of RCS to specify an exhaustive list of supported access mechanisms.

As described in section 2.8 both trusted and untrusted networks need to be able to provide access and authentication over NAT. This must be taken into account for example when xSIM based access, IPSec or other fixed access authentication mechanisms are used.

This support for access over non-cellular networks can be used in two ways:

1. As an offloading capability for the cellular network
   This will be controlled by the device itself:
   - When a device is enabled for VoLTE/VoHSPA (see section 2.2.1), it is expected that the device remain on LTE/HSPA access as long as it is available.
   - When the voice service is provided via CS access, it is up to the device when and whether to move to non-cellular (e.g., Wi-Fi) access. If a device moves to a non-cellular network, it is expected that the device first de-registers in IMS from the cellular network, and then registers in IMS in the non-cellular network, or vice versa when moving in the other direction.
2. As a means of access for dedicated broadband clients using the identity of the mobile device

This can be either as a standalone client when there is no mobile device using that same identity or as secondary client to a mobile device sharing the same identity (see chapter 2.5). In the latter case the user will have multiple devices sharing the same identity. Chapter 2.11 provides further details on how this can be realized. These differences are further detailed in section 2.9.2.2.

2.9.2.2 Dedicated RCS-AA clients on Broadband Access

Next to clients using mobile access, RCS also supports dedicated clients using broadband access. Such a client can operate in two significantly different modes:

1. As a secondary client, adding performance (such as larger keyboard, a screen with higher resolution and so on) to the primary mobile client with RCS functionality. Such a secondary client is designed with user experience aspects, storage accessibility and so on, but is not designed to act as a primary telephony device. In this case the primary client retains aspects a user would associate with their device, for example regulatory functions, quality of service and full access to the telephony functions.

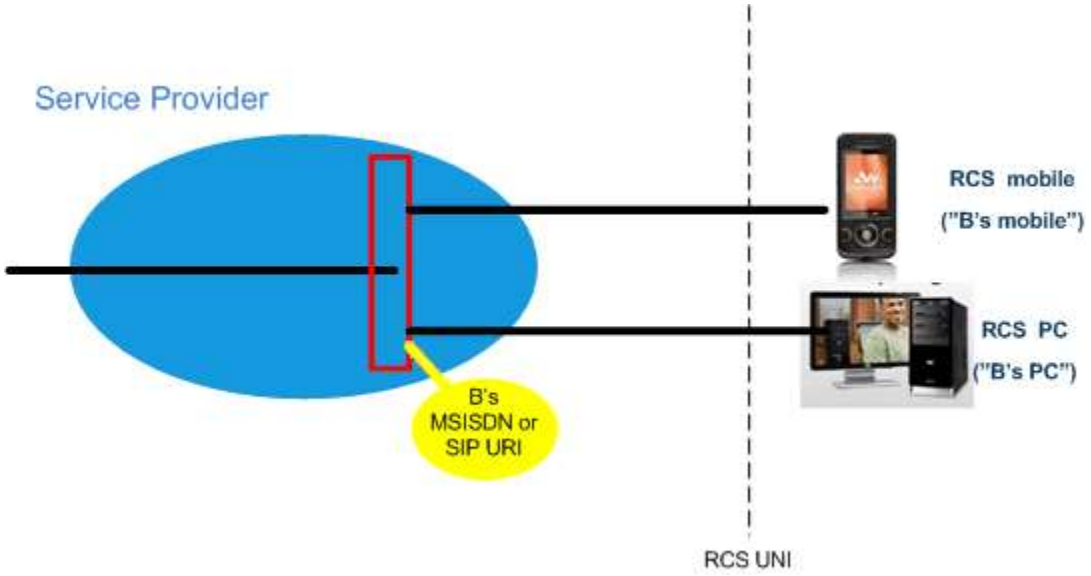


Figure 34: RCS Broadband Access client used as a secondary client

NOTE: Other combinations of multiple devices, such as support of multiple mobile clients, are out of scope for RCS. However, this does not restrict a Service Provider to deploy proprietary solutions to achieve this.

2. As a primary client, replacing the user's mobile client. A primary client has to meet all regulatory requirements (emergency calling, lawful intercept, etc.), and perform to meet the traditionally expected telephony functionality and demonstrate the reliability, performance and quality of service of a primary device. The precondition for its use is that basic telephony services are already available in the Broadband Access network. For these services, the local regulations are already fulfilled.

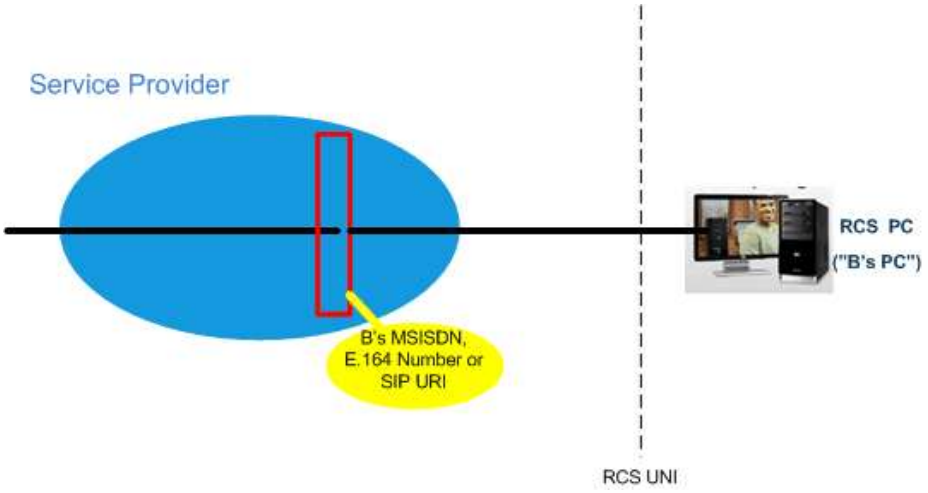


Figure 35: RCS Broadband Access client used as a primary client

2.10 End User Confirmation Requests

The following section provides a framework that will allow the Service Provider to inform the end user about a certain situation by opening a dialog in the device presenting all the available information and asking the user to confirm or decline the proposed request.

The End User Confirmation Request is implemented using a SIP MESSAGE[23] method containing a XML payload type "*application/end-user-confirmation-request+xml*" that will be sent by the Service Provider serving the end user to his RCS device/client. A specific device can be addressed using a GRUU or a sip.instance feature tag (see section 2.11.3). If the user is required to answer from every device, the devices should be addressed individually using a GRUU or a sip.instance feature tag.

Upon the reception of the SIP MESSAGE, the end user terminal will check the *P-Asserted-Identity* of the incoming message and match it against the configured URI for the service (END USER CONF REQ ID) as defined in Table 93 and extract the request information from the XML payload body. A dialog or notification will be displayed to the End User (UX dependent) showing the confirmation request and related information.

The End User Confirmation Response will be encapsulated in an XML body with a payload type "*application/end-user-confirmation-response+xml*" and returned back to the Service Provider in a new SIP MESSAGE.

[23] Please take into account that according to [RFC3428], the size of MESSAGE requests outside of a media session MUST NOT exceed 1300 bytes, unless the UAC has positive knowledge that the message will not traverse a congestion-unsafe link at any hop, or that the message size is at least 200 bytes less than the lowest MTU (Maximum Transmission Unit) value found en route to the UAS. Larger payloads may be sent by the Service Provider in the initial confirmation request and/or ack (Acknowledgement) using content-indirection as specified in [RFC4483]. Therefore, this shall be supported by the devices/clients.

2.10.1 End User Confirmation Request

The information contained in the end user confirmation request is the following

- **Id**: Unique identifier of the request.
- **Type**: Determines the behaviour of the receiving device. It can take one of the following two values:
    a) *Volatile*, the answer shall be returned inside of a new SIP MESSAGE request. The request may time out without end user input, in which case it will be discarded.
    b) *Persistent*, the answer shall be returned inside of a new SIP MESSAGE request. The confirmation request does not time out.
- **Pin**: Determines whether a pin is requested to the end user. It can take one of the following two values: *true* or *false*. If the attribute is not present it shall be considered as *false*. This pin request can be used to add a higher degree of confirmation and can be used to allow certain operations like parental control for example.
- **Subject**: text to be displayed as notification or dialog title.
- **Text**: text to be displayed as body of the dialog.
- **Timeout**: Time period in seconds during which a volatile request is valid. After the timeout expires, the device shall discard any UX notifications silently.

For volatile type requests an optional timeout attribute may be present in the XML representing the validity period in seconds. If this attribute is not present a default value of 64*T1 seconds (with T1 as defined in [RFC3261]) shall be used.

The End User Confirmation Request initiates a dialogue to the user on the device. For specific use cases it may be necessary that the user accepts external End User Confirmation Requests which cannot be authenticated appropriately. This acceptance can either be done by configuration or by entering a specific mode on the device UI and avoids unwanted UI dialogues on the devices caused by malicious usage by other person. One use case described in sections 2.3.3.4.2.2 and 2.3.3.4.2.3 is the configuration of additional RCS clients via Internet. To identify such messages following attribute is provided:

- externalEUCR: Determines that this is an End User Confirmation Request initiated by an external unsecure source, e.g. via the Internet. If the optional attribute externalEUCR is set to true in the End User Confirmation Request and the device does not allow such external End User Confirmation Request, the End User Confirmation Request shall be ignored. An End User Confirmation Request response shall not be sent back in that case. The device shall show all End User Confirmation Request requests where the attribute externalEUCR is set to false or does not exist.
  If the device or client has not implemented the processing of the attribute externalEUCR, it shall be ignored and therefore all End User Confirmation Requests are allowed and shown in the UI.

In addition, to allow Service Providers more flexibility the two following optional button labels will be defined. For backward compatibility: if the optional button labels are not used, default values will be used instead.

- **ButtonAccept**: text to display on the button.
- **ButtonReject**: text to display on the button.

To ensure compatibility with future versions, the RCS client/device shall silently discard any unknown node or attribute in the XML structure.

Several Subject or Text nodes can be present in the XML body to be able to support multiple languages. If more than one element is presented a language (*lang*) attribute must be present with the two letter language codes according to the ISO 639-1. RCS clients shall

verify the language attribute and display the text data of the element that matches the current language used by the user. If there is no language matching the users, the first node of Subject and Text shall be used.

If the type of confirmation request is persistent the Service Provider can send an optional acknowledgement message of the transaction back to the user with a welcome message, an error message or further instructions. This acknowledgement message will be encapsulated in an XML body with a payload type "*application/end-user-confirmation-ack+xml*" and returned in a separate SIP MESSAGE. If the acknowledgement refers to the message which is currently displayed, it shall be discarded even if no answer was sent. This allows sending a message to all active devices of a user also when a response from a single device is sufficient. For that reason it is also possible to send acknowledgements without Subject or textual content.

The following table specifies the XML Schema Definition (XSD) of the XML payload for the End User Confirmation Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns:xml=”http://www.w3.org/XML/1998/namespace”
            elementFormDefault="qualified">
      <xs:import namespace=”http://www.w3.org/XML/1998/namespace”
                  schemaLocation="http://www.w3.org/2009/01/xml.xsd"/>
      <xs:element name="EndUserConfirmationRequest">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="Subject" maxOccurs="unbounded"/>
                        <xs:element ref="Text" maxOccurs="unbounded"/>
                        <xs:element ref="ButtonAccept" minOccurs="0" maxOccurs="unbounded"/>
                        <xs:element ref="ButtonReject" minOccurs="0" maxOccurs="unbounded"/>
                  </xs:sequence>
                  <xs:attribute name="id" type="xs:string" use="required"/>
                  <xs:attribute name="type" type="xs:string" use="required"/>
                  <xs:attribute name="pin" type="xs:boolean" use="optional"/>
                  <xs:attribute name="timeout" type="xs:integer" use="optional"/>
                  <xs:attribute name="externalEUCR" type="xs:boolean" use="optional"/>
            </xs:complexType>
      </xs:element>
      <xs:element name="Subject">
            <xs:complexType>
                  <xs:simpleContent>
                        <xs:extension base=”xs:string”>
                              <xs:attribute ref="xml:lang"/>
                        </xs:extension>
                  </xs:simpleContent>
            </xs:complexType>
      </xs:element>
      <xs:element name="Text">
            <xs:complexType>
                  <xs:simpleContent>
                        <xs:extension base=”xs:string”>
                              <xs:attribute ref="xml:lang"/>
                        </xs:extension>
                  </xs:simpleContent>
            </xs:complexType>
      </xs:element>
      <xs:element name="ButtonAccept">
            <xs:complexType>
```

```
                    <xs:simpleContent>
                        <xs:extension base=”xs:string”>
                            <xs:attribute ref="xml:lang"/>
                        </xs:extension>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
            <xs:element name="ButtonReject">
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:extension base=”xs:string”>
                            <xs:attribute ref="xml:lang"/>
                        </xs:extension>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
</xs:schema>
```

Table 46 : End User Confirmation Request XSD

2.10.2 End User Confirmation Response

The information contained in the End User Confirmation Response is the following:

- **Id**: Unique identifier of the request.
- **Value**: with the end user confirmation. It can take one of the following two values accept or decline.
- **Pin**: if the request has the “*pin*” attribute set to true, the response will contain the pin value introduced by the user.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns:xml=”http://www.w3.org/XML/1998/namespace”
            elementFormDefault="qualified">
        <xs:import namespace=”http://www.w3.org/XML/1998/namespace”
                    schemaLocation="http://www.w3.org/2009/01/xml.xsd"/>
        <xs:element name="EndUserConfirmationResponse">
            <xs:complexType>
                <xs:attribute name="id" type="xs:string" use="required"/>
                <xs:attribute name="value" type="xs:string" use="required"/>
                <xs:attribute name="pin" type="xs:string" use="optional"/>
            </xs:complexType>
        </xs:element>
</xs:schema>
```

Table 47: End User Confirmation Response XSD

The information contained in the End User Acknowledge Response is the following

- **Id**: Unique identifier of the original request. If the ID matches the ID of the currently shown message, this message shall be discarded even if no answer was sent from the receiving device.
- **Status**: of the End User Confirmation. It can take one of the following two values: *ok* or *error*.
- **Subject**: text to be displayed as notification or dialog title
- **Text**: text to be displayed as body of the dialog.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
                xmlns:xml="http://www.w3.org/XML/1998/namespace"
                elementFormDefault="qualified">
        <xs:import namespace="http://www.w3.org/XML/1998/namespace"
                        schemaLocation="http://www.w3.org/2009/01/xml.xsd"/>
        <xs:element name="EndUserConfirmationAck">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="Subject" maxOccurs="unbounded"/>
                                <xs:element ref="Text" maxOccurs="unbounded"/>
                        </xs:sequence>
                        <xs:attribute name="id" type="xs:string" use="required"/>
                        <xs:attribute name="status" type="xs:string" use="required"/>
                </xs:complexType>
        </xs:element>
        <xs:element name="Subject">
                <xs:complexType>
                        <xs:simpleContent>
                                <xs:extension base="xs:string">
                                        <xs:attribute ref="xml:lang"/>
                                </xs:extension>
                        </xs:simpleContent>
                </xs:complexType>
        </xs:element>
        <xs:element name="Text">
                <xs:complexType>
                        <xs:simpleContent>
                                <xs:extension base="xs:string">
                                        <xs:attribute ref="xml:lang"/>
                                </xs:extension>
                        </xs:simpleContent>
                </xs:complexType>
        </xs:element>
</xs:schema>
```

Table 48: End User Confirmation Acknowledgement XSD

2.10.3 End User Notification Request

To provide more flexibility a Service Provider shall be able to send only notification messages to the end user. This notification message shall be implemented similar to confirmation dialog using a SIP MESSAGE method containing an XML payload type "*application/end-user-notification-request+xml*". A notification will be displayed to the end user (UX dependent) showing the related information.

The information contained in the end user notification is the following:

- **Id**: Unique identifier of the request.
- **Subject**: text to be displayed as notification or dialog title
- **Text**: text to be displayed as body of the dialog.
- **ButtonOK**: text to display on the button.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns:xml=”http://www.w3.org/XML/1998/namespace”
            elementFormDefault="qualified">
      <xs:import namespace=”http://www.w3.org/XML/1998/namespace”
                  schemaLocation="http://www.w3.org/2009/01/xml.xsd"/>
      <xs:element name="EndUserNotification">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="Subject" maxOccurs="unbounded"/>
                        <xs:element ref="Text" maxOccurs="unbounded"/>
                        <xs:element ref="ButtonOK" minOccurs="0" maxOccurs="unbounded"/>
                  </xs:sequence>
                  <xs:attribute name="id" type="xs:string" use="required"/>
            </xs:complexType>
      </xs:element>
      <xs:element name="Subject">
            <xs:complexType>
                  <xs:simpleContent>
                        <xs:extension base=”xs:string”>
                              <xs:attribute ref="xml:lang"/>
                        </xs:extension>
                  </xs:simpleContent>
            </xs:complexType>
      </xs:element>
      <xs:element name="Text">
            <xs:complexType>
                  <xs:simpleContent>
                        <xs:extension base=”xs:string”>
                              <xs:attribute ref="xml:lang"/>
                        </xs:extension>
                  </xs:simpleContent>
            </xs:complexType>
      </xs:element>
      <xs:element name="ButtonOK">
            <xs:complexType>
                  <xs:simpleContent>
                        <xs:extension base=”xs:string”>
                              <xs:attribute ref="xml:lang"/>
                        </xs:extension>
                  </xs:simpleContent>
            </xs:complexType>
      </xs:element>
</xs:schema>
```

Table 49: End User Notification XSD

2.10.4 End User System Request

It shall be also possible to send System Request to the RCS client to trigger an internal action based on the type of the request. These requests are not displayed to the user at the UI level. The request is implemented also using a SIP MESSAGE method containing an XML payload body of type “*application/system-request+xml*”.

The information contained in the end user notification is the following:

- **Id**: Unique identifier of the request.
- **Type**: Identifying the kind of action to be triggered

- **Data**: Custom information needed to perform the action.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
	<xs:element name="SystemRequest">
		<xs:complexType>
			<xs:attribute name="id" type="xs:string" use="required"/>
			<xs:attribute name="type" type="xs:string" use="required"/>
			<xs:attribute name="data" type="xs:string" use="optional"/>
		</xs:complexType>
	</xs:element>
</xs:schema>
```

Table 50: System Request XSD

The following list shows the defined system requests in this RCS specification:

Type	Data	Action
urn:gsma:rcs:http-configuration:reconfigure	N.A.	Perform an HTTP reconfiguration. See section 2.3.5.3
urn:gsma:rcs:extension:control	List of (<IARI>,<duration>) separated by ';'	Prevent each Extension from the list identified by their IARIs to access the RCS infrastructure for the duration provided (in seconds). A duration of 0 indicates that the Extension shall be permanently prevented from using the RCS infrastructure. A <IARI> is formatted in the following way: *urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.ext.<identifier>* example: *(urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.ext.A5TgS99bJloIUIh1209SJ82B21m87S1B87SBqfS871BS8787SBXBA3P45wjp63tk,0);(urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.ext.VoxgS94bJloIUIh12r9Sop1j21m87Spt83SZqfS871BS128pSB2B13P42wjp43rt,3600)*

Table 51: List of System Requests in RCS

2.10.5 Example Use Case 1: Accepting terms and conditions

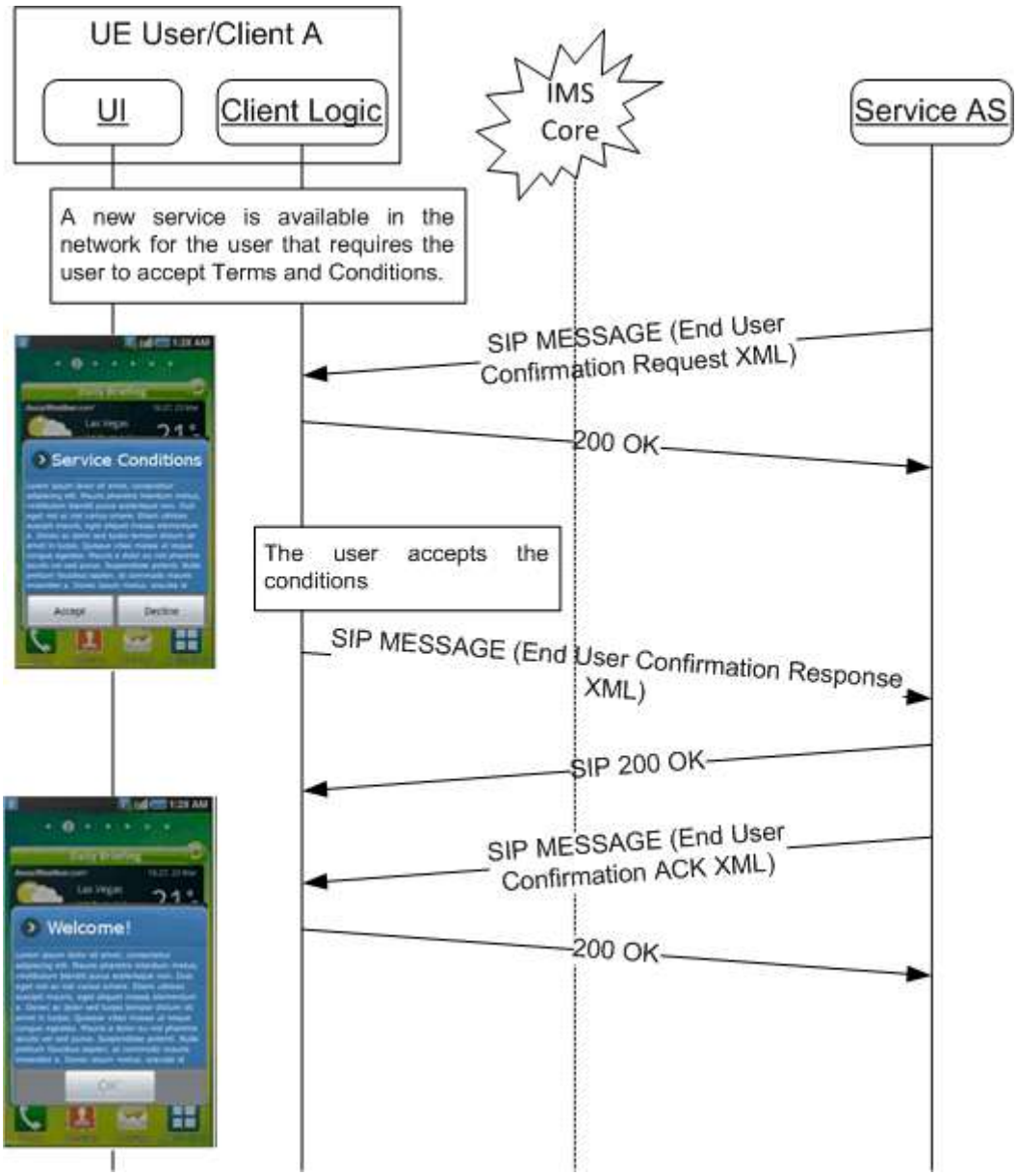


Figure 36 : Terms and Condition Use Case example

2.10.6 Example Use Case 2: Notification

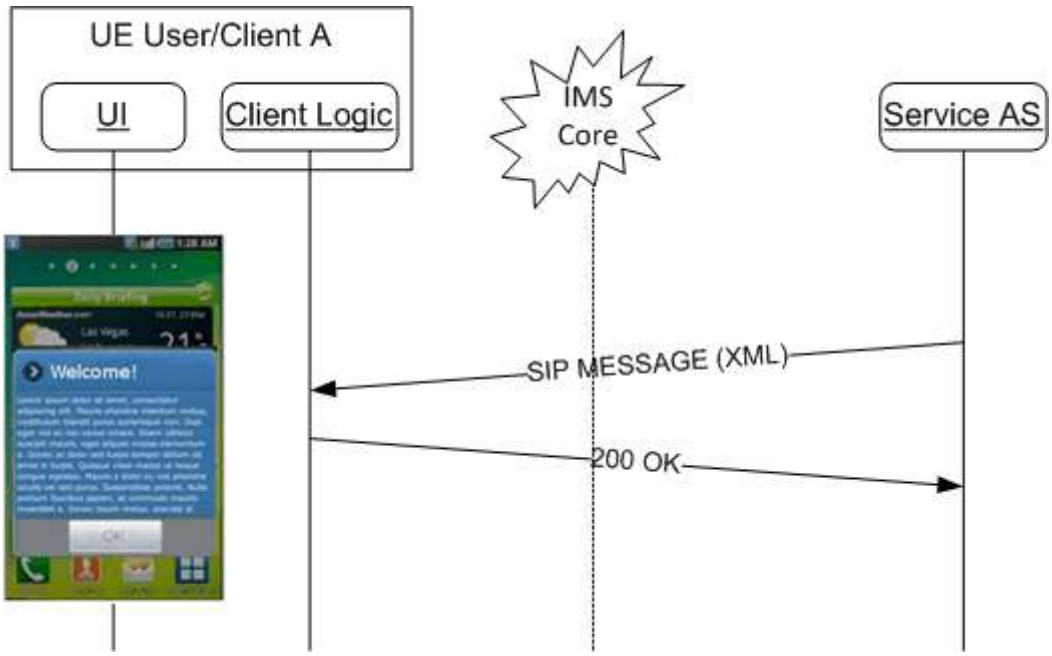


Figure 37: User Notification example

2.11 Multidevice support

2.11.1 Overview

As shown in section 2.9.2.2, the use of a broadband access client leads to the possibility of the user having multiple devices that share the same (public) identity, a MSISDN for instance. Depending on the services that are deployed, this multidevice environment allows a user to:

- Answer a call or respond to a message from a device/client that suits their purpose
- Have a single buddy list shared between the devices/clients
- Authorize invitations to share Social Presence Information from every device/client
- Have a single Social Presence Information that can be seen and maintained from every device/client that is used

The general communication behaviour in this environment is that when the recipient has multiple devices/clients in use and a call or a message is received every recipient's device will alert. The recipient may then respond to the call or to the message from any of their devices; whichever device is the best for the current situation. In addition when the recipient accepts or rejects a call from any of the devices, all the other devices will stop alerting.

To achieve this, an RCS client shall send a SIP 603 DECLINE response to the invite request when an RCS User explicitly declines a session invitation for a SIP session based service like for example an IP Voice Call, File Transfer, Video and Image Share. According to [3GPP TS 24.229] and [RFC3261] both such a rejection and an acceptance will result in a SIP CANCEL request sent by the S-CSCF to the other devices of the user that have not yet accepted nor rejected the invitation. In both cases, the requests may carry a Reason header field as specified in [RFC3326] that is populated with the proper SIP response code values (as per [3GPP TS 24.229]), in this case either the *cause=200* or *cause=603* values.

If the user device has accepted the INVITE with a 200 OK, then the S-CSCF should set the Reason header field with *SIP* protocol and the protocol-cause set to *200* along with an optional protocol-text (e.g. *SIP;cause=200;text="Call completed elsewhere"*).

In case one device has sent a 603 “Declined” then the S-CSCF should set the *cause=603* along with an optional protocol-text (e.g. *SIP;cause=603;text="Declined"*), in either SIP CANCEL and/or SIP BYE, towards the remaining user devices.

When a client receives a SIP CANCEL request containing a Reason Header field with the protocol set to “SIP” and the protocol-cause set to 200, a client may for example use this information to indicate to the user that the session was accepted on another device (rather than as for example a missed call).

As a fallback for legacy services where this general communication behaviour cannot be realised a call or message might be directed to a certain device.

2.11.2 Control of Service delivery

This feature is applicable to secondary clients only and does not affect the primary device that will always receive all supported communications that it’s capable of given the circumstances.

This feature gives the user an option to control the flow of communication. In some cases a user may not be willing to answer calls from a secondary device (for example a PC). The difference between muting and control of service delivery is that the control of service delivery:

- Disables the service: the user no longer receives calls, messages or requests for the service
- When the service is disabled the user cannot use the service to make calls, send messages or requests

To provide a good user experience, the device/client must clearly show that the service/services are disabled in the device/client user interface.

As a default setting the secondary device will receive all the communication that it can support. Whether the settings are maintained after a restart of the client when the user had changed them, is out of scope of this document is the choice of the device/client vendor.

When an end-user decides that they do not want to use a certain service on a secondary client, that client shall reject any incoming requests related to that service, with SIP 486 BUSY HERE without alerting the end-user. Furthermore, the client would not offer the possibility to use that particular service.

The control of service delivery can be offered for following services:

- Voice Calls
- Video Calls
- Chat
- Text Messaging
- Multimedia Messaging
- File Transfer
- Video Sharing
- Image Sharing
- Geolocation PUSH

The actual set of services on which this control of service delivery will be offered to the end user may be a subset of the above list. Which service is/is not part of that subset shall be determined by the client capabilities and a Service Provider controllable parameter (See Annex A).

2.11.3 Addressing of individual clients

If a client obtains GRUUs from the registrar as described in section 2.4, the public GRUU shall be used as device identifier. The client shall use the public GRUU as a URI parameter for the client in non-REGISTER requests and responses that it sends, for example, an INVITE request and 200 OK response where the GRUU will be included in the Contact Header.

If a client does not obtain a GRUU from the registrar, the sip.instance feature tag and value shall be used as the device identifier. The client shall include the sip.instance feature tag in the Contact header with the same instance-id value in any non-REGISTER request and responses that it sends.

NOTE: A scenario when the network does not support GRUU and the client type is embedded and has access to the IMEI, causes a privacy issue since the device will send a plain IMEI in all SIP requests/responses thus revealing the IMEI to remote end.

For delivery and display notifications for chat messages which are expected to be directed to the original sender of a message, the client builds the SIP request to send the notification using the received device identifier.

For a Standalone Pager Mode Message (i.e., via SIP MESSAGE) requesting a delivery or display notification, the identity of the sender is populated in the CPIM From header field and can include one of the following (see examples in section 3.4.4.1.8):

- the public GRUU of the sender's client, or
- the authenticated sender's IMS public identity plus the sip.instance value for the sender's client.

If an *IMDN-Record-Route* header was received in the corresponding chat message, the recipient client shall include in the Request-URI the topmost entry from the *IMDN-Route* header.

Otherwise if no *IMDN-Record-Route* header (see [RFC5438]) was received in the CPIM wrapper (see [RFC3862]) of the message, then if the original SIP INVITE request associated with the message included the GRUU parameter in its Contact header, or if the original SIP MESSAGE request carrying a Standalone Pager Mode Message included the GRUU parameter in the CPIM From header field, the recipient client adds the GRUU as the Request-URI when it builds the SIP request to send the notification as described in section 5.2.5.2 of [RCS5-SIMPLEIM-ENDORS] or in section 6.1.1 of [RCS5-CPM-CONVFUNC-ENDORS]. In all cases if a GRUU was received, the recipient client shall set the CPIM *To* header in the notification to the public GRUU of the received SIP INVITE request.

If the original SIP INVITE request associated with the message had no GRUU parameter, but did have a *sip.instance* feature tag in the *Contact* header, or if the original SIP MESSAGE request carrying a Standalone Pager Mode Message had no GRUU parameter in the CPIM From header field but did have the sip.instance value in the CPIM From header field the recipient client sends the notification with a separate *Accept-Contact* header to carry the *sip.instance* feature tag and its value. The client shall include the *explicit* and *require* tags on that header.

If the original SIP INVITE request delivering the chat messages contained a *Conversation-ID* header (as described in [RCS5-CPM-CONVFUNC-ENDORS]), the recipient client should include the *Conversation-ID* with the same value and should include a *Contribution-ID* header (as described in [RCS5-CPM-CONVFUNC-ENDORS]) with a newly generated value.

For standalone messages, similar procedures are followed as explained in sections 7.2.4.1 and 7.2.4.2 of [RCS5-CPM-CONVFUNC-ENDORS].

For file-transfer service, in case of file resume, the file recipient needs to address the device that sends the file initially. If the file transfer is resumed from the file sending side, the file sending client needs to address the device of the file recipient that has accepted the original file transfer. If present in the initial SIP INVITE, the device identifier (sip.instance) or GRUU shall be used accordingly (for details, see section 3.5.4).

For simplicity and given that the long-term approach using GRUU as defined in [RFC5627] is preferred, the diagrams contained in Annex B show the network supporting *pub-gruu*. The diagrams for a network supporting the *sip.instance* tag only, would be equivalent except for changing the mechanism to carry the device identifier (*sip.instance* instead *pub-gruu*).

2.11.4 Routing RCS SIP requests to RCS Clients

In the context of multi-client and multi-device deployments, it is possible that the same IMS public user identity (IMPU) is used by the RCS Client and by other IMS Clients (e.g. Voice over LTE client using SMS over IP), but carrying a different instance ID.

In the messaging case, to ensure that requests are forked only to RCS Clients which have explicitly registered with the required RCS capabilities, the terminating Messaging Server may add a dedicated Accept-Contact header field to each RCS SIP request carrying either a CPM ICSI, or a SIMPLE IM feature tag, with the *require* and *explicit* parameters described in [RFC3841]. This applies to the following services:

- Chat
- Standalone Messaging (text and multimedia messaging)
- File Transfer

2.12 Interconnect principles and guidelines

The Service Provider's IMS NNI shall follow the provisions in [PRD-IR.65] sections 3, 4, 5 and 6.

The Service Provider's RCS NNI shall follow the provisions in [PRD-IR.90]. The implementation could be any of the three connectivity options for RCS NNI defined in [PRD-IR.90].

The Service Provider's RCS interconnection shall follow the provisions in [PRD-AA.60] including the service specific annexes.

2.13 Security and privacy

2.13.1 Access Security for the User-to-Network Interface (UNI)

2.13.1.1 Access Signalling Security Methods

Several SIP signalling access security and authentication methods are specified in [3GPP TS 33.203] and [3GPP TS 24.229] for access to the IMS core and IMS based services such as RCS. The applicability and choice of method is highly dependent on the RCS client and access type (e.g. trusted or untrusted) including what is supported or required by the IMS core.

2.13.1.1.1 GPRS IMS Bundled Authentication (GIBA)

GIBA is an "early" IMS access authentication method for access over a GSM/GPRS or UMTS network whereby the underlying PS domain is providing the access security and authentication on behalf of the IMS core. In this scenario, the Gateway GPRS Support Node (GGSN) allocates a PDP (Packet Data Protocol) context for the mobile device and in doing so the assigned IP address, along with the IMSI and MSISDN is sent to a RADIUS server/Home Subscriber Server (HSS) for the PS domain. Authentication to the IMS core is

done by ensuring that the IP address (policed by both the GGSN and P-CSCF) and the IMS identities received in SIP signalling correspond to those allocated for the mobile's PDP context in the PS domain.

GIBA is generally applicable for GPRS or UMTS access from mobile devices which do not support AKA based xSIM credentials or for devices or IMS core networks which do not support "*ipsec-3gpp*" established using the SIP Security Agreement (*sec-agree*) and IMS AKA as specified in [3GPP TS 33.203].

2.13.1.1.2 IMS AKA with IPsec

IMS AKA with IPsec is the preferred long term approach in IMS for access signalling security from a cellular PS network. Such access requires the IMS client device to possess an AKA based credential (e.g. Universal SIM (USIM)/IP Multimedia Services SIM (ISIM)) and support the "*ipsec-3gpp*" procedures specified in [3GPP TS 33.203] and [3GPP TS 24.229].

IMS AKA with IPsec is the access signalling approach specified for Voice over LTE (VoLTE) ([PRD-IR.92]).

2.13.1.1.3 SIP Digest Authentication and TLS

SIP Digest is a username and password challenge based authentication (based on HTTP Digest) which is suited for broadband access to IMS or for RCS clients which do not possess AKA based credentials (e.g. xSIM) or do not support IMS AKA based IPsec. SIP Digest is widely implemented in Internet Engineering Task Force (IETF) based SIP clients and is often deployed with TLS. Support for SIP Digest with and without TLS is specified in [3GPP TS 33.203] and [3GPP TS 24.229] for access to IMS from "non-3gpp" defined access networks (e.g. broadband/fixed access networks).

When an RCS client is enabled for SIP Digest authentication, the client will use the pre-configured SIP username and password as specified in Table 88 to authenticate to the IMS core. For the initial SIP REGISTER message (before a digest challenge) the RCS client shall include an authorization header (as per [3GPP TS 24.229]) which includes the SIP digest username and an empty digest authentication response parameter. This allows the IMS core to treat the SIP digest username as an IMS private user identity (IMPI) which is distinct from the IMS public user identity (IMPU), allowing the same SIP public user identity (or IMPU) to be registered from multiple RCS clients/devices.

The IMS registration flow for SIP digest authentication is shown in Figure 38. In this example flow, the RCS client is connected to the IMS core over a Wi-Fi internet broadband connection.

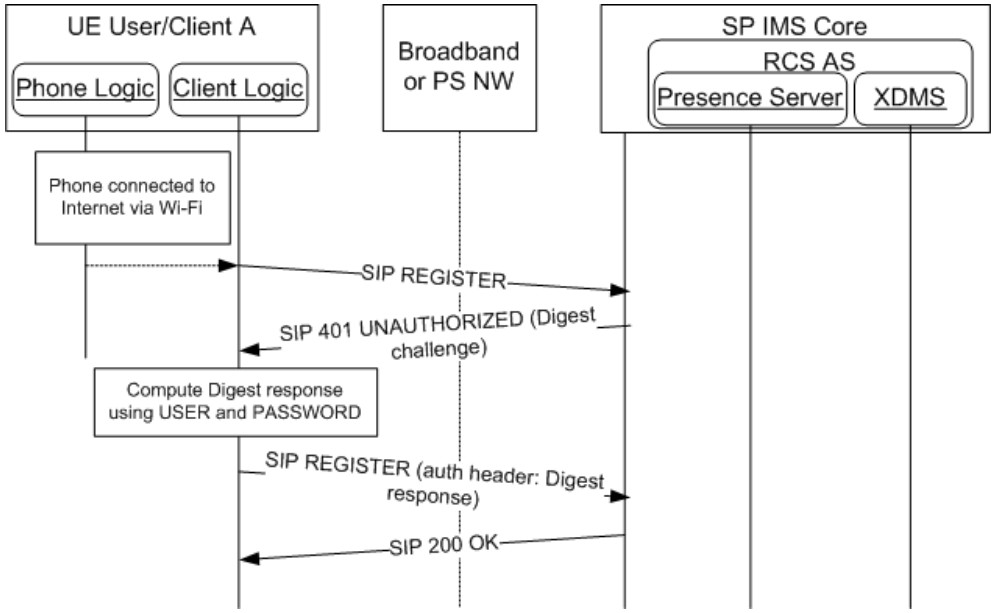


Figure 38: Registration with SIP Digest Authentication

If digest authentication fails two times with a SIP 401 UNAUTHORIZED response, the client shall not attempt further registration attempts, but rather consider the current configuration as invalid and force a reconfiguration using the procedures in chapter 2.3 the next time the handset is rebooted.

The use of SIP Digest with TLS is recommended for access from untrusted access networks (including WLAN with no encryption). TLS provides per message authentication, integrity protection and encryption for SIP signalling. TLS with server side certificates also provides authentication of the IMS core to the RCS client. Note that this requires the client to possess a root or intermediate certificate of a Certificate Authority (CA) that is in the certificate signing chain for the IMS core's (e.g. P-CSCF) TLS certificate.

When an RCS client is enabled to use SIP/TLS it should use the SIP TLS port obtained through P-CSCF discovery procedures (e.g. through DNS SRV records [Service records]) or configuration. However, if RCS client is not able to determine a SIP TLS port through these means, it shall use the default SIP port for TLS as specified in [RFC3261].

The RCS client enabled to use SIP/TLS should first use the SIP security agreement (sec-agree) [RFC3329] as specified in [3GPP TS 24.229] to first negotiate the use of TLS with its SIP Proxy (P-CSCF). Alternatively an RCS client may first try to establish a TLS session with the SIP proxy (P-CSCF) before sending an initial SIP Register message which does not include sec-agree for TLS. However, with this approach the S-CSCF may challenge subsequent non-Register messages with a 407 Proxy Authentication Required unless configured to trust SIP Digest without signalling security indicated or if the P-CSCF is able to provide this indication despite not using sec-agree.

Note in both cases SIP proxy (P-CSCF) authenticates to the RCS client using a TLS server certificate.

When SIP Digest is not used with TLS, the IMS core may require non-REGISTER SIP requests to be authenticated using the same SIP Digest challenge mechanisms used during registration. However, in this case the SIP digest challenge is sent in a 407 (Proxy Authentication Required) response. An RCS client that receives a 407 (Proxy Authentication Required) response shall respond by sending an authenticated SIP request which includes a Proxy Authorization header with the digest response. The RCS client shall cache the digest challenge data (e.g. server nonce) for use in authenticating subsequent SIP requests using a nonce-count value (for replay protection) as per [RFC2617] and including a Proxy

Authorization header with an updated digest response. This avoids the need for the IMS core to challenge each SIP request before the authentication data expires. Once the digest authentication data expires a new challenge will be issued.

NOTE: the IMS core may also support binding the RCS client's IMS identities authenticated during registration with a source IP address (and port if [RFC5626] "SIP Outbound" is used). In such cases, the IMS core may not require subsequent non-registration based SIP messaging to be authenticated using SIP Digest if the identities and source addresses in the messaging matches the binding obtained during the Digest authenticated registration process.

2.13.1.2 Access Signalling Security Profiles for RCS

As there are several considerations which access signalling security method should be used for access to RCS services, the following table defines authentication and access signalling security mechanisms as per RCS device and access type.

Device	Access	Applicable Security Methods	Applicability and Suitability
Mobile client not configured for VoLTE/VoHSPA (RCS-CS mode)	Cellular PS Access	GIBA or SIP Digest (with or without TLS) or IMS AKA with IPsec	GIBA (e.g. SSO) applies only to GPRS and UMTS access for mobile devices IMS AKA with IPSec may be used when supported by both device and the network. SIP Digest with or without TLS is used in cases when pre-configured or where GIBA is pre-configured, but not supported by the network
	Non-cellular broadband (Wi-Fi) access	SIP Digest, SIP Digest with TLS or IMS AKA with IPsec	SIP Digest with TLS is recommended over SIP Digest without TLS SIP Digest with or without TLS is used in cases when pre-configured or where GIBA is pre-configured or when the mobile device does not support IMS AKA for WLAN access
VoLTE/VoHSPA configured mobile client (RCS-VoLTE, RCS-VoHSPA, RCS-CS modes possible)	Cellular PS Access	IMS AKA with IPsec[24] Note that the configuration to any other method is not possible.	AKA credentials stored securely in a UICC such as an xSIM.
	Non-cellular broadband (Wi-Fi) access	SIP Digest, SIP Digest with TLS or IMS AKA with IPsec[24].	SIP Digest with TLS is recommended over SIP Digest without TLS SIP Digest with or without TLS is used in cases when pre-configured or where GIBA is pre-configured or when the mobile device does not support IMS AKA for WLAN access.
Broadband Access Enabled (RCS-AA mode)		SIP Digest or SIP Digest with TLS	SIP Digest with TLS is recommended over SIP Digest without TLS SIP Digest is used for mobile devices which do not support IMS AKA for WLAN access.

Table 52: Access Signalling Security Profiles for RCS

[24] Requires UDP encapsulation of IPsec for NAT traversal

For RCS devices which can access the IMS core from both mobile and broadband/fixed networks (e.g. Wi-Fi) a separate access signalling security method and corresponding authentication credential may be required. If the security mechanism is not pre-configured as per section A.1.6.3 and A.2.10, the RCS device negotiates the set of security mechanisms using the SIP security agreement [RFC3329] as specified for IMS in [3GPP TS 33.203] and [3GPP TS 24.229]. If the client is pre-configured with a specific access signalling security mechanism, the client uses the signalling corresponding to this security method in the initial registration procedure, and the IMS core determines (based on signalling) which mechanism is being used/requested and then determines (based on security policy) if the access signalling security method is allowed.

NOTE: the RCS device shall support a configuration option for each of these profiles (where applicable).

For those cases where the GIBA is pre-configured, but the client supports also SIP Digest, behaviour shall be as follows:

- SSO/GIBA authentication takes place first
- If it fails (e.g. Service Provider network equipment does not support it) digest authentication is then tried

2.13.1.3 Access Media Security

2.13.1.3.1 Secure RTP (SRTP)

SRTP [RFC3711] may be used to provide per message authentication, integrity protection and encryption for both RTP and RTCP streams involved in real-time video and voice sessions. The use of SRTP is recommended for communications over any untrusted network in which confidentiality (or lack of) is a concern. As an example, a voice or video call over a Wi-Fi network (e.g. “Hot Spot”) without any WLAN (Wireless Local Area Network) encryption is highly susceptible to eavesdropping.

The establishment and key exchange for SRTP in RCS shall be based on SDES (Session Description Protocol Security Descriptions for Media Streams, cf. [RFC4568]) which is transported within SDP, following the SIP SDP offer/answer model. SDES and SRTP profiles for media security in IMS are specified in [3GPP TS 33.328].

Note that [3GPP TS 33.328] defines two modes of operation for SDES/SRTP: e2ae (end-to-access edge) mode and e2e (end-to-end) mode. For the e2ae mode, SDES is run between an IMS client and a SIP edge proxy, i.e. a P-CSCF (IMS-ALG). An IMS access Gateway controlled by a P-CSCF (IMS-ALG [Application Layer Gateway]) provides the SRTP termination for the “Access Edge”. In the e2e mode, SDES and SRTP is transported end-end between two end user clients.

An RCS client that supports SRTP and SDES and support e2ae mode shall indicate this during the IMS registration according to [3GPP TS 24.229]. The P-CSCF (IMS ALG), if supporting e2ae mode, indicates this to the UE as part of the IMS registration procedures according to [3GPP TS 24.229]. The use of SRTP is enabled through the client configuration parameters (see section A.2.10), and whether it is used or not can be configured differently for Wi-Fi access and cellular access.

However not all end user clients may support SRTP. Therefore the Service Provider’s network equipment should support e2ae mode. An RCS client that supports SRTP and SDES shall also support e2ae mode.

When using SRTP/SDES, the RCS client can include preference of security mode to use in accordance to [3GPP TS 33.328]. It is recommended that e2ae mode is used by the UE, if also indicated to be supported by the P-CSCF (IMS-ALG). Otherwise, the RCS client may try e2e by not indicating any preference during the session setup. Note that this does not

exclude that the Service Provider network still may decide to terminate the media security in the network (P-CSCF (IMS-ALG)).

For terminating sessions, when the UE has indicated support for e2ae SRTP/SDES in the registration, the P-CSCF (IMS-ALG) shall behave as specified in [3GPP TS 24.229], i.e., ensure that SRTP is used, and facilitate interworking from RTP to SRTP when needed.

For terminating session, when the UE has not indicated support for e2ae SRTP/SDES, the P-CSCF (IMS ALG) decides based on local policy, whether to apply SRTP / SDES towards the UE. A possible local policy is that the P-CSCF (IMS-ALG) invokes procedures related to SDP and SRTP for Wi-Fi access, but not for cellular access.

Note that enforcing SRTP/SDES on the terminating call leg towards a UE that does not support SRTP/SDES will lead to the connection establishment failing, which may be an issue for inbound roaming where the operator has no control of what clients are used, or for cases where there are other (non-RCS) clients in the same network that use RTP.

2.13.1.3.2 MSRP

MSRP is used in many RCS services which involve the exchange of images, files and instant messages (e.g. session based). Similar to RTP, MSRP is established through SDP exchanges in SIP signalling and it relies heavily on the security provided in signalling. The use of cryptographically strong random values appended to MSRP URIs exchanged within SDP provides binding between the SIP and MSRP sessions and any identities exchanged within SIP.

For RCS, the use of TLS mode as specified in [RFC4975] is recommended when MSRP is transported over an unsecure network (e.g. Wi-Fi). Consequently, a client configuration parameter to enable Message Session Relay Protocol over Transport Layer Security (MSRPoTLS) is specified in section A.2.10, and whether it is used or not can be configured differently for Wi-Fi access and cellular access.

The RCS client shall use self-signed TLS certificates to produce fingerprints (e.g. secure hash) of the certificate which are exchanged during the SDP negotiation associated with the invitation and MSRP establishment procedure. The certificate fingerprint used for MSRP shall follow the same fingerprint mechanism specified in [RFC4572]. This binding of the certificate fingerprint to SIP signalling relies on the underlying security and trust provided by SIP signalling (e.g. IPsec, SIPoTLS (SIP over TLS), etc.). As a consequence, it is assumed that MSRPoTLS connections shall only happen when combined with the use of encrypted SIP signalling.

When using MSRPoTLS, and with the following two objectives allow compliance with legal interception procedures, the TLS authentication shall be based on self-signed certificates and the MSRP encrypted connection shall be terminated in an element of the Service Provider network providing service to that UE. Mutual authentication shall be applied as defined in [RFC4572].

Since the alternative connection model for MSRP shall be supported as specified in [RFC6135] (see section 2.8) the network will in some cases take the active role, and in some cases take the passive role, in the establishment of the TCP connection. Each peer (UE and network) shall take the same role (active or passive) in TLS as it took in TCP, so if the network has taken the passive role in TCP, it will also act as TLS server, as specified in [RFC6135]. When TLS is used, both endpoints shall exchange self-signed TLS certificates and fingerprints, as specified in [RFC4572].

In RCS, and in accordance with [RFC4975], all UEs are mandated to support MSRPoTLS as defined in [3GPP TS 24.229-rel12] with certificate fingerprints as defined in [3GPP TS 33.328]. For terminating sessions, the P-CSCF (IMS ALG) decides based on local policy whether to apply MSRPoTLS towards the UE. A possible local policy is that the P-CSCF

(IMS-ALG) invokes procedures related to MSRPoTLS for Wi-Fi access, but not for cellular access.

2.13.1.4 XCAP Authentication and Security

XML Configuration Access Protocol (XCAP) exchanges between the RCS client and the XDMS requires authentication and in most cases transport layer security.

Authentication may be provided through the use of HTTP Digest authentication and may use the same credential (e.g. username and password) as SIP based Digest authentication when applicable. For RCS clients that use IMS AKA based credentials for SIP access (e.g. VoLTE), a separate credential may be required unless the IMS Generic Authentication Architecture (GAA) is supported, along with the procedures in [3GPP TS 33.222] for obtaining a suitable credential (e.g. HTTP digest secret) for HTTPS based access such as for XCAP. In GAA, the IMS AKA credential is used in a “bootstrapping” process to obtain other types of credentials from the “bootstrapping” server.

For RCS clients enabled for VoLTE/VoHSPA (see section 2.2.1), the same authentication specified for VoLTE use of XCAP as defined in [PRD-IR.92] or [PRD-IR.58] shall be supported as follows:

- For RCS clients (and IMS core) that support GAA, the RCS client shall use their AKA credential to fetch an HTTP digest credential using the 3GPP Generic Bootstrapping Architecture (GBA). The RCS client authenticates to an Authentication Proxy (AP) over an HTTP/TLS (HTTPS) secured session using HTTP Digest as per [RFC2617].
- For RCS clients (and IMS core) that do not support GAA, the RCS client shall use its pre-configured credential (e.g. username and password) to authenticate to the AP over an HTTP/TLS (HTTPS) secured session using HTTP Digest as per [RFC2617].

For non-VoLTE/VoHSPA enabled RCS clients, the use of HTTP authentication ([RFC2617]) over an HTTP/TLS (HTTPS) secured session shall be supported for XCAP authentication to the XDMS (or AP).

The HTTP digest credentials (e.g. username and password) for XCAP are specified in section A.1.2.

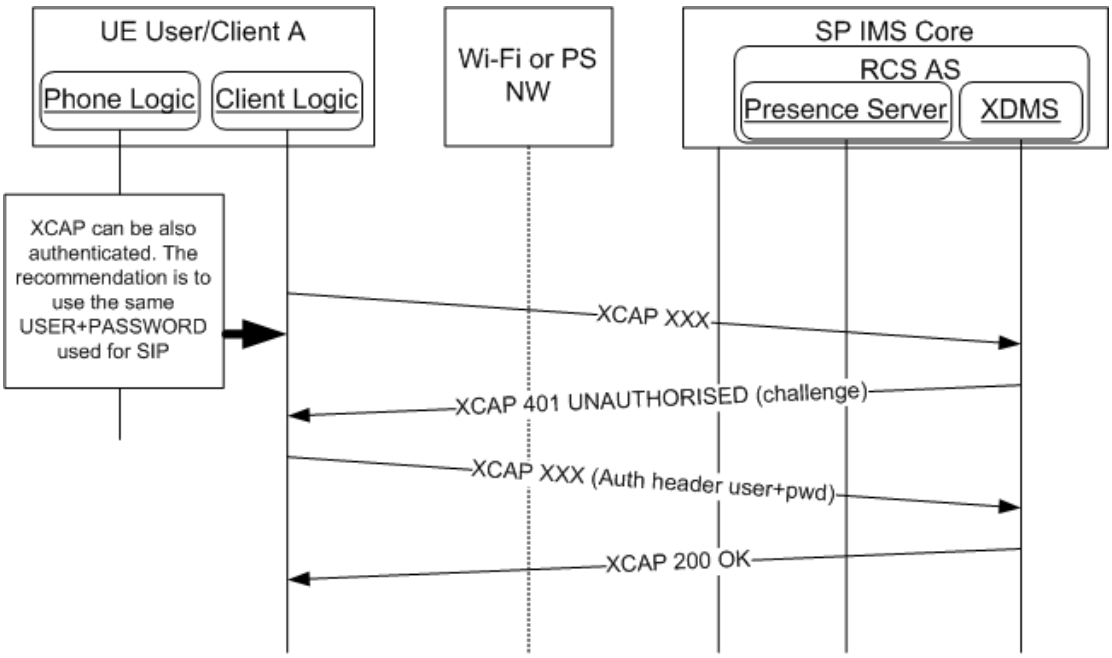


Figure 39: XCAP authentication using HTTP Digest

In the case of “Early IMS Security” in which GIBA is used for SIP authentication to the IMS core, the use of a credential such as a username and password (or IMS AKA) may not be required to authenticate to the XDMS (or an AP). A similar IP address based authentication approach as GIBA is specified in [3GPP TS 33.141] Annex D for access to HTTP based services using Early IMS Security. Support for this mechanism requires the XDMS or an AP to support the procedures in [3GPP TS 33.141] Annex D to fetch the IP address binding of the RCS client from the HSS, using the “*X-3GPP-Intended-Identity*” header provided by the client.

2.13.1.5 Message Content Store Authentication and Security

The RCS client shall support the authentication and security mechanisms described in [RCS5-CPM-MSGSTOR-ENDORS] for access to the Message Content Store using IMAP.

Authentication shall be based on username and password stored on the RCS client and one of the following IMAP authentication methods:

- Plaintext username and password sent using the LOGIN command as specified in [RFC3501]
- Simple Authentication and Security Layer (SASL) based mechanism using the AUTHENTICATE command as specified in [RFC3501].

For the SASL based authentication, the “PLAIN” SASL authentication method shall be used as specified in [RFC3501] and [RFC2595].

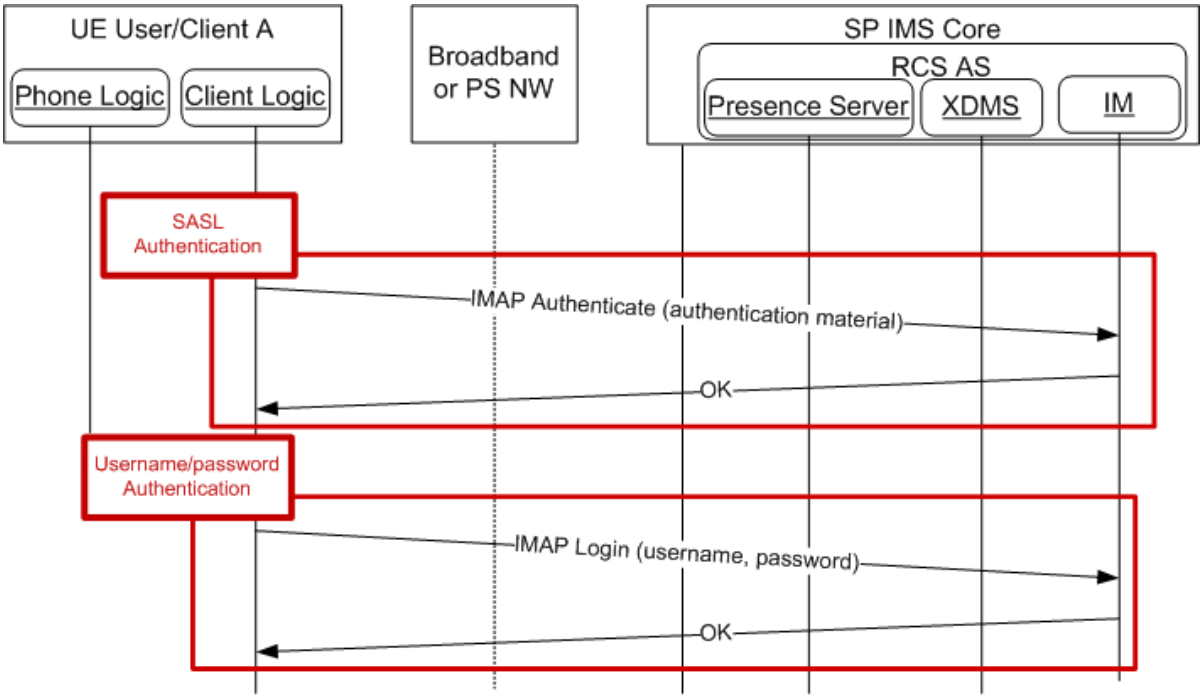


Figure 40: IMAP authentication with SASL or plain text login

TLS shall be used to provide message authentication, integrity protection and confidentiality for the IMAP protocol as specified in [RCS5-CPM-MSGSTOR-ENDORS]. TLS must first be established using the STARTTLS command before any IMAP based authentication occurs using either the LOGIN or AUTHENTICATE command.

The Message Content Store server shall authenticate itself towards the RCS client using certificate based TLS authentication. The client shall support certificates based on a Public Key Infrastructure (PKI) for which the RCS client is pre-configured with a root or intermediate CA (which is recommended to be a public CA root authority) certificate in the signing chain of the certificate.

2.13.2 Privacy

2.13.2.1 Overview

A key element of promoting user adoption of RCS is gaining the user's trust with regards to privacy. Service Providers need to provide security mechanisms to ensure unwanted parties cannot gain access to RCS user communications and provide adequate mechanisms to enable users to control the information they share. The key security measures to meet these requirements are outlined in section 2.13.1 and privacy controls are summarised in section 2.13.2.2.

2.13.2.2 Privacy controls

Mechanisms provided in RCS to enable users to control their privacy are identified in this section.

2.13.2.2.1 Multidevice Privacy

Where an RCS user has RCS active across multiple devices this fact shall be obscured from other users.

NOTE: Where an RCS user has RCS active across multiple devices this fact cannot be obscured from devices of other users, since the GRUU and/or sip.instance feature tags shall automatically indicate this fact to these other devices.

2.13.2.2.2 Presence information Privacy

The RCS user shall have the option of controlling who they share their presence information with through a process of accepting, blocking or ignoring an invitation to establish a presence relationship (see section 3.7.4.5).

2.13.2.2.3 Video Privacy

The RCS client shall provide the RCS user with control over when any camera on the device is active.

2.13.2.2.4 Social Presence Information Privacy

The RCS user shall have the option to disable sharing of social presence information.

2.13.2.2.5 Network Address Book Privacy

The Service Provider shall ensure access control to the Network Address Book via a process of authentication.

2.13.2.2.6 Location Privacy

The RCS user shall have the option to control sharing of location information (see section 3.10.1.2).

2.13.2.2.7 Messaging and Chat

An RCS user shall have the option to control information communicated about their actions during messaging communications and chat sessions, including the suppression of "display" notifications and "IsComposing" notifications.

2.14 XDM Handling and Shared XDMS

The support of XDM is an optional functionality for RCS and is only required for Service Providers deploying one of the following services:

- The capability discovery based on presence (see section 2.6.1.2)
- The Personal Network Blacklist (see section 2.15)
- Social Presence (see section 3.7)

2.14.1 Shared XDMS template

Following template shall be used for the *resource-lists* in the Shared XDMS:

Shared XDMS:
AUID: resource-lists
Document name: index
Template:

```
<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
                xmlns:xd="urn:oma:xml:xdm:xcap-directory"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <!-- The list oma_buddylist contains references to any individual list used according to OMA IG for presence
   subscriptions. -->
   <list name="oma_buddylist">
      <external anchor="http://xcap.gsma.org/resource-
      lists/users/sip:1234578901@gsma.org/index/~~/resource-lists/list%5B@name=%22rcs%22%5D"/>
      <external anchor="http://xcap.gsma.org/resource-
      lists/users/sip:1234578901@gsma.org/index/~~/resource-
      lists/list%5B@name=%22rcs_basic_spi_only%22%5D"/>
   </list>

   <!-- The list rcs_poll_buddylist contains references to individual lists used for RCS non-VIP Contacts -->
   <list name="rcs_poll_buddylist">
      <external anchor="http://xcap.gsma.org/resource-
      lists/users/sip:1234578901@gsma.org/index/~~/resource-lists/list%5B@name=%22rcs_poll%22%5D"/>
      <external anchor="http://xcap.gsma.org/resource-
      lists/users/sip:1234578901@gsma.org/index/~~/resource-
      lists/list%5B@name=%22rcs_poll_basic_spi_only%22%5D"/>
   </list>

   <!-- The list oma_grantedcontacts contains the list of all granted contacts -->
   <list name="oma_grantedcontacts">
      <external anchor="http://xcap.gsma.org/resource-
      lists/users/sip:1234578901@gsma.org/index/~~/resource-lists/list%5B@name=%22rcs%22%5D"/>
      <external anchor="http://xcap.gsma.org/resource-
      lists/users/sip:1234578901@gsma.org/index/~~/resource-lists/list%5B@name=%22rcs_poll%22%5D"/>
   </list>

   <!-- The list rcs_basic_spi_only_grantedcontacts contains the list of all basic SPI Only granted contacts -->
   <list name="rcs_basic_spi_only_grantedcontacts">
      <external anchor="http://xcap.gsma.org/resource-
      lists/users/sip:1234578901@gsma.org/index/~~/resource-
      lists/list%5B@name=%22rcs_basic_spi_only%22%5D"/>
      <external anchor="http://xcap.gsma.org/resource-
      lists/users/sip:1234578901@gsma.org/index/~~/resource-
      lists/list%5B@name=%22rcs_poll_basic_spi_only%22%5D"/>
   </list>
```

```
<!-- The list oma_blockedcontacts contains the list of all blocked contacts. -->
<list name="oma_blockedcontacts">
    <external anchor="http://xcap.gsma.org/resource-
    lists/users/sip:1234578901@gsma.org/index/~~/resource-
    lists/list%5B@name=%22rcs_blockedcontacts%22%5D"/>
    <external anchor="http://xcap.gsma.org/resource-
    lists/users/sip:1234578901@gsma.org/index/~~/resource-
    lists/list%5B@name=%22rcs_revokedcontacts%22%5D"/>
</list>

<!-- The list of VIP contacts (buddies) the owner wants to provide all social presence information to. This list
also includes the owner's own URI -->
<list name="rcs">
    <display-name>My presence buddies with location sharing</display-name>
    <entry uri="tel:+1234578901"/>
</list>

<!-- The list of VIP Contacts (buddies) the owner wants to provide only basic social presence information to --
>
<list name="rcs_basic_spi_only">
    <display-name>My presence buddies without location sharing</display-name>
</list>

<!-- The list of NON-VIP Contacts (buddies) the owner wants to provide all social presence information to -->
<list name="rcs_poll">
    <display-name>My NON-VIP presence contacts with location sharing</display-name>
</list>

<!-- The list of NON-VIP Contacts (buddies) the owner wants to provide only basic social presence
information to -->
<list name="rcs_poll_basic_spi_only">
    <display-name>My NON-VIP presence contacts without location sharing</display-name>
 </list>

<!-- The list of blocked contacts -->
<list name="rcs_blockedcontacts">
    <display-name>My blocked contacts</display-name>
</list>

<!-- The list of revoked contacts -->
<list name="rcs_revokedcontacts">
    <display-name>My revoked contacts</display-name>
    <entry uri="tel:+123456" xd:last-modified="2008-12-24T14:32:14Z"/>
</list>
<list name="rcs_pnb_chat_blockedusers">
    <display-name>My chat blacklist</display-name>
</list>

<list name=" rcs_pnb_ft_blockedusers">
    <display-name> My file transfer blacklist </display-name>
</list>

<list name=" rcs_pnb_standalone_blockedusers">
    <display-name> My standalone blacklist </display-name>
</list>

<list name=" rcs_pnb_outchat_blockedusers">
```

```
        <display-name>My outgoing chat blacklist</display-name>
    </list>

    <list name=" rcs_pnb_outft_blockedusers">
        <display-name> My outgoing file transfer blacklist </display-name>
    </list>

    <list name=" rcs_pnb_outstandalone_blockedusers">
        <display-name>My standalone IM blacklist</display-name>
    </list>
</resource-lists>
```

Table 53: Shared Lists template for RCS

NOTE1: the entry in the "*rcs_revokedcontacts*" list is for illustrative purposes only. It is included as an example since it deviates slightly from the standard list usage. The entry in the "*rcs*" list is also for illustrative purposes only, showing that the user's own URI will be included so the user's clients receive the user's own presence information (see also section 3.7.4.3.3).

NOTE2: the resource-list contains only the lists needed for the features allowed by the service provider (e.g. all the lists related to presence should not be added if only PNB is deployed).

2.14.2 XML Document Handling

When first started the RCS client shall check through a XCAP directory query whether

- The "resource-lists" document exists, if Presence or PNB is deployed by the service provider.
- The "pres-rules", "rls-services" and the "pidf-manipulation" (permanent presence state) documents exist, if Presence is deployed by the service provider.

If they do not exist, the RCS client shall create them if they are applicable (i.e. depending on whether Presence or PNB is deployed as described in the previous bullets). If the documents exist, the RCS client will check whether they comply with the templates defined in sections 2.14.1 and 3.7.4.5.2 by using the following criteria for the documents:

- For the "*resource-lists*" document, first check whether it contains an "*rcs_basic_spi_only*" list. If not, add the "*rcs_basic_spi_only*" and "*rcs_basic_spi_only_grantedcontacts*" lists to the document and modify the "*oma_buddylist*" list to refer to both the "*rcs*" and the "*rcs_basic_spi_only*" lists.
- Secondly check whether it contains an "*rcs_poll_buddylist*" or an "*rcs_poll*" list. If not, add the "*rcs_poll*", "*rcs_poll_basic_spi_only*" and "*rcs_poll_buddylist*" lists to the document and modify the "*oma_grantedcontacts*" list to refer to both the "*rcs*" and the "*rcs_poll*" lists and the "*rcs_basic_spi_only_grantedcontacts*" list to refer to both the "*rcs_basic_spi_only*" and "*rcs_poll_basic_spi_only*" lists.
- For the "*rls-services*" document, firstly check if the "*rcs*" service URI entry refers to the "*oma_buddylist*" list. If the document refers to the "*rcs*" list instead, the RCS client shall modify it to refer to the "*oma_buddylist*" list
- Secondly, check if it contains an "*rcs_poll*" service URI entry. If not, an "*rcs_poll*" service URI entry with a reference to the "*rcs_poll_buddylist*" in Shared XDMS will be added.
- For the "*pres-rules*" document, check whether it contains the "*rcs_basic_spi_only_granted_contacts*" rule. If not, the RCS client shall add this rule to the document.

- For "*rcs_pnb_chat_blockedusers*", "*rcs_pnb_ft_blockedusers*" "*rcs_pnb_standalone_blockedusers*", "*rcs_pnb_outchat_blockedusers*", "*rcs_pnb_outft_blockedusers*" and "*rcs_pnb_outstandalone_blockedusers*" lists in the Shared XDMS, if they do not exist and the PNB MANAGEMENT configuration parameter (see section A.1.2) is set to enabled, they shall be added to the "*resource-lists*" document.

Once the documents have been setup in this way, the RCS client shall only modify the "*rcs*", "*rcs_basic_spi_only*", "*rcs_poll*", "*rcs_poll_basic_spi_only*", "*rcs_revokedcontacts*" and "*rcs_blockedcontacts*", "*rcs_pnb_chat_blockedusers*", "*rcs_pnb_ft_blockedusers*", "*rcs_pnb_standalone_blockedusers*", "*rcs_pnb_outchat_blockedusers*", "*rcs_pnb_outft_blockedusers*" and "*rcs_pnb_outstandalone_blockedusers*lists in the "*resource-lists*" document. Only if the user explicitly requests to recreate the documents according to the possibility described below, the other documents and parts of the "*resource-lists*" document should be modified.

XDM documents can be updated without the involvement of the RCS client of this RCS release. Two types of changes are possible:

1. Shared lists are updated by adding new entries, removing entries or updating entries.
2. Structural changes to the documents (for example to support new options in the presence authorization).

In case 1, in order not to overwrite changes done for example by another client, either a conditional update should be done (per XCAP conditional operations as defined in [RFC4825] section 7.11) or the client should retrieve the latest status of the document before doing the update. An RCS client of this RCS release shall support one of these options when updating XDM documents.

Case 2 (structural changes to a XDM document) could occur when an RCS client of this RCS version is deployed in a future RCS environment, even though the future RCS version should be backward compatible with previous ones. The RCS client shall go to a read-only mode with regards to all XDM documents when it detects such changes. Future RCS versions will indicate this by renaming the "*rcs*" shared list. If the list is not renamed, but structural changes were detected in documents in the presence and RLS XDMS, the RCS client will go to read-only mode only for the updated documents. In that case the RCS client indicates to the user that they should use a client with an updated RCS version to carry out commands that require modifying any of such documents.

Circumstances where the user downgrades from a future RCS release to the use of an RCS client only, (for example the end-user does not have a client with an updated RCS version or there is some blocked situation between the XDMC and XDMS), the RCS client shall offer the user the possibility to remove all information stored in the XDMS's, this then creates new documents based on its current status and RCS release. The removal of the documents shall be based on a retrieval of the complete list of documents using XCAP Directory requests and then removing all listed documents (thus including documents unknown to the RCS client of this RCS release) using relevant operation such as XCAP PUT/DELETE.

Should a device for its own internal use maintain a local copy of the Shared XDMS's "*resource-lists*" document (see section 2.14.1) or the information contained therein, then it shall verify with the Shared XDMS whether its copy is still up to date in the following situations:

- When the client comes online
- When it receives a notification within the dialog of its RLS subscription indicating that the subscription to a contact is pending or active and according to the locally maintained information, it is not aware that the user is part of the RCS buddy list.

NOTE1: this situation can occur, when the user invites the contact to share social presence information from another client, or a contact has been added as a VIP-contact from another client.

- When it receives a notification within the dialog of its watcher information subscription indicating that a subscription from a contact changed from the "pending" to the "*active*" or "*terminated*" state when no action was taken to authorize or block that subscription from the client. The state change to "*terminated*" should only be taken into account for this case when the event triggering the state change indicates "*rejected*".

  NOTE2: this situation can occur when the user authorizes or blocks the subscription from another client.

- When it receives a notification within the dialog of its RLS subscription indicating that the subscription to a contact that is presence enabled was terminated with reason "*timeout*" when no action was taken from the client to revoke the presence sharing with that contact.
- When it receives a notification within the dialog of its RLS subscription indicating that the subscription to a contact that is presence enabled was terminated with reason "*noresource*" when no action was taken from the client with that contact.

  NOTE3: this situation can occur, when the user changes a contact from being a VIP contact to being a non-VIP contact from another client.

  NOTE4: a device is not required to maintain a local copy of the Shared XDMS's "*resource-lists*" document. If it does not, for presence it can simply display the presence information it receives and it does not need to access the XDMS.

2.14.2.1 Client XML procedures and multi-device

The XML Document Management Client (XDMC) from the RCS client performs the XCAP Get and Put operations on the Shared List XDMS *resource-lists*. Once updated by the user, the RCS client shall store the updated version of the document in the Shared List XDMS for further access from his other devices.

The XDMC shall cache the '*resource-lists*' document locally and subscribe to the updates of that document as described in [XDM2.0_Core] when such subscriptions are enabled through the XDM CHANGES SUBSCRIPTION parameter described in section A.1.2 of Annex A. If this is the case, all user's RCS clients that have subscribed to the 'resource-lists' document changes will receive a SIP NOTIFY whenever the content of any of the lists has changed. If subscriptions are not enabled, the RCS client shall not cache the document and fetch the latest version of the document from the XDMS prior to displaying the list to the user or enabling them to make modifications in it.

2.14.2.2 Authorizing XCAP Requests

XCAP requests need to be authorized by the XDMS. This authorization relies on an assertion of the identity of the requestor of an XCAP request.

The HTTP header fields *X-XCAP-Asserted-Identity* and *X-3GPP-Asserted-Identity* used to contain the asserted identity of a requestor of an XCAP request may depend on operational conditions (type of access used by the terminal, Service Provider policy) for example different Service Providers may apply different algorithms to assert the identity of a requestor of an XCAP request. Thus, for any Authorization check to be carried out by the XDMS, any of both *X-XCAP-Asserted-Identity* and *X-3GPP-Asserted-Identity* header fields

are accepted as a valid header field containing the asserted identity of the requestor of the XCAP request inside the Service Provider domain.

To offer a unique inter-Service Provider interface, the *X-3GPP-Asserted-Identity* header field is always conveyed between two Service Provider domains, at the NNI interface.

When the terminal of a watcher requests, via XCAP, some content (for example status-icon, refer to section 3.7.4.4.2.3) associated with the presence document of a presentity, the XDMS of the presentity has to check whether the watcher is authorized to access this content, according to the presentity's presence subscription rules.

As defined in sections 2.14.1 and 3.7.4.5.2, amongst others the "*rcs*" list is granted this permission.

The lists in section 2.14.1 can contain both SIP URI and tel URI address of authorized watchers in a Service Provider domain. To ensure both cases at the NNI interface, the "*X-3GPP-Asserted-Identity*" of the initiator of an XCAP request should contain both the sip URI and tel URI of this user.

2.15 Personal Network Blacklists (PNB)

With this optional RCS feature that is enabled using the PNB MANAGEMENT configuration parameter (see section A.1.2), the RCS user may be provided with the possibility to manage their Personal Network Blacklist (PNB), in order to either prevent receiving undesired communications, messages or media.

The PNB feature relies on the Shared XDMS that is also used for SPI (see section 3.7). New lists are pre-defined in the Shared List XDMS as follows:

- "*rcs_pnb_chat_blockedusers*": this list contains all blocked senders for chat
- "*rcs_pnb_ft_blockedusers*": this list contains all blocked senders for file transfer
- "*rcs_pnb_standalone_blockedusers*": this list contains all blocked senders for standalone messages
- "*rcs_pnb_outchat_blockedusers*": this list contains all blocked recipients for chat
- "*rcs_pnb_outft_blockedusers*": this list contains all blocked recipients for file transfer
- "*rcs_pnb_outstandalone_blockedusers*": this list contains all blocked recipients for standalone messages

2.15.1 RCS Applicability

The enforcement of the PNB feature is performed by the Blacklist Policy Enforcement Function (BPEF) that could both:

- Be implemented as part of the relevant RCS application server (e.g. RCS Messaging Server), or,
- In a separate server, which enforces the policy.

The PNB feature applies only to the following RCS services:

- Standalone messaging (see section 3.2)
- 1-to-1 chat (see section 3.3)
- File transfer (see section 3.5).

Taking into account the supported features, the BPEF can be completely collocated with the Messaging Server.

The PNB feature can be enabled or disabled. When enabled:

- The BPEF shall apply triggers for checking the RCS user's PNB, during the relevant RCS service traffic, on both originating and terminating traffic.

- RCS clients shall be configured with the *PNB MANAGEMENT* parameter (described in Annex A, section A.1.2), so it is possible to handle the PNB configuration from the client.

The functionality provided by BPEF is summarized below:

- On originating side:
  - Checks the outgoing blacklists for the respective request (i.e. chat, file transfer or standalone messaging) and if the recipient is found in the applicable list (e.g. rcs_pnb_outchat_blockedusers for chat), the request is rejected.
- On the terminating side:
  - Checks the blacklists for the respective request (i.e. chat, file transfer or standalone messaging) and if the sender is found on the pertaining list (e.g. chat_blockedusers for chat), the request is rejected.
  - If supporting the Common Message Store feature and if allowed by local server policy, the Messaging Server stores the blocked message/chat/File Transfer event with the metadata content following Conversation History format as per [RCS5-CPM-MSGSTOR-ENDORS].

2.15.2 PNB management

The PNB can be updated in any of the following ways:

1. The user performs the management of the lists from their RCS client by adding or deleting users in the list entries of the Shared List XDMS using XCAP, or
2. The RCS client supports the user, by prompting him/her to choose upon a rejection (e.g. chat, group chat or file transfer) to add the originator in the respective blacklist.

Once the user rejected a Chat invitation, or a File Transfer, the RCS client UI may prompt them to select whether the rejection is permanent, or if it is a onetime rejection. A permanent rejection will trigger an update in the user's respective black list, by adding the originator in that list.

Other options may be possible, such as allowing the RCS user to add other users in their PNBs from the viewing of the Conversation History.

Regardless of the trigger for the PNB update by the RCS Client, when such update needs to be done in the network, the RCS Client shall issue an XCAP request as per [XDM2.0_Core] procedures and as described in section 2.14.2.

2.15.2.1 Authorizing PNB management requests

The authorization of the PNB management requests (get, update, delete) is done as described in section 2.14.2.2.

2.16 Emergency Services

2.16.1 General

In some markets, regulatory requirements are emerging for IMS Multimedia Emergency Services. UEs and the network in required markets must support the 3GPP Release 11 IMS Emergency Services as specified in [3GPP TS 24.229-rel11], [3GPP TS 23.167], Chapter 6 and Annex H, and 3GPP Release 11 emergency procedures specified in [3GPP TS 24.301].

Please note [PRD-IR.92] and [PRD-IR.58] in Section 5.2 specify Emergency Services support.

2.16.2 RCS Service Feature List

Emergency Services support is provided in the following RCS Service Feature:

- 1-to-1 Chat

3 RCS 5 Services

3.1 General Service Overview

RCS provides several services that fit into the framework defined in section 2. As mentioned in section 1.2 all of these services are optional for a Service Provider to deploy.

The first set of services is intended to enhance the user's messaging experience. Section 3.2 describes the standalone messaging service based on OMA CPM that is considered as an evolution of the SMS/MMS messaging services providing fewer restrictions and provides the interworking capability with those services. Section 3.3 introduces the 1-to-1 chat service that provides a more real-time experience through "IsComposing" indications next to the store and forward functionality, including delivery and display notifications, that allows reaching users while they are offline. In section 3.4 it is described how the 1-to-1 chat service is extended to multiparty scenarios. For both the 1-to-1 chat and for this Group Chat, the technical realization can be based on either OMA SIMPLE IM or OMA CPM. Interworking between these realisations has been described to manage these as a single service providing transparency and an enhancement to the UX.

As a service that is closely related to the messaging in that it is used for the exchange of discrete content and is based on the same underlying technology, chapter 3.5 describes the File Transfer service allowing a user to exchange any type of file with another user.

Chapter 3.6 introduces the content sharing services allowing the user to exchange a video or image in real-time with another user. For video sharing this can be done both within a call and outside of a call, while the sharing of images is only available during a call. In other circumstances the File Transfer service could be used.

The social presence service in chapter 3.7 allows the user to announce a status including a picture, a link and possibly even information related to his location to a subset of his contacts while at the same time receiving status updates from those same contacts. Depending on the user's preference regarding a contact, they could be informed about such status changes in real-time or after a potentially long delay.

Section 3.8 and 3.9 describe respectively an IP based voice and video call functionality for broadband access clients and mobile devices on HSPA and LTE. These services include support for a set of supplementary services and ensure the quality of service delivery when used on HSPA and LTE access. For the voice call, a mobile device on HSPA and LTE provides continuity to a CS call if network coverage circumstances require this. These services are based on [PRD-IR.58] and [PRD-IR.92] for the voice call and [PRD-IR.94] for the video call.

A geolocation service is introduced in section 3.10 which allows a user to share their location (or any other desired location) with a contact including requesting the location of a contact.

All these services can be invoked either from within the address book provided that the contact has the corresponding capability (see section 2.6) and the current network connectivity allows using the service (see section 2.6.4.1) or directly from the device's menu. Additional entry points may be the chat and call history, the media gallery and camera application depending on what is suitable for the service.

Most of the NNI handling is done as described in section 2.12.

3.2 Standalone messaging

RCS provides a Standalone Messaging service as described in [RCS5-CPM-CONVFUNC-ENDORS]. It includes both text and multi-media messaging services using IMS-based OMA CPM Standalone Messaging instead of the SMS and the MMS.

The use of OMA CPM Standalone Messaging removes some of the limitations associated with a messaging service deployment based on the SMS and MMS services, e.g., the 160-character message size, content type, lack of display notifications for text messages and support for the service users with multiple devices.

In addition, the RCS Standalone Messaging service supports interworking to SMS and MMS as described in [RCS5-CPM-IW-ENDORS].

A conversational view of the CPM standalone messaging is used in RCS, and it can be synchronized between a user's multiple devices, by making use of the CPM Common Message Store as described in [RCS5-CPM-MSGSTOR-ENDORS].

3.2.1 Feature description

The feature list of the RCS standalone messaging service includes the following main features:

- Standalone messaging (text and multimedia)
- Delivery and Display Notifications
- Support for multiple devices per user
- Deferred Messaging
- Common Message Store
- Interworking with legacy messaging services

These features are further described in sections below.

3.2.1.1 Standalone messaging

The RCS standalone messaging capability employs the OMA CPM's SIP-based standalone messaging as described in [RCS5-CPM-CONVFUNC-ENDORS]. It evolves the two separate text and multimedia messaging mechanisms into one single and unified messaging framework. This converged messaging mechanism uses the combination of the Pager Mode messaging mechanism and the Large Message Mode messaging mechanism. The mode is selected based on the message size. Smaller messages are sent via Pager Mode and larger messages via Large Message Mode. This built-in capability of the RCS Standalone Messaging enhances the user experience by making the selection transparent to the user: the user does not have to choose between messaging technologies based on either the media type or artificially imposed size limits. In addition the RCS Standalone Messaging further facilitates the transition from the currently distinct SMS and MMS messaging services towards a single all-IP Messaging services.

The RCS standalone messaging includes support for the following specific features:

1. In supporting both text and multi-media messaging, it does not make a distinction between text and multimedia messages.
2. Its message delivery includes both 1-to-1 and group messaging including support for "reply-to-all" functionality.
3. Imposes no limitations on the message size and media types. However, the maximum message size can be controlled by Service Providers.
4. Capabilities for both broadband access and mobile access terminals.
5. It can store a message exchange both in a local and a Common Message Store and to present a conversational view of the exchanged messages.
6. Provides message delivery and display notifications.

3.2.1.2 Delivery and display notifications

Upon sending an RCS Standalone Message including a request for message disposition state, the sender shall receive a delivery notification and may receive a display notification.

If an RCS Standalone Message contains a disposition notification request targeted at a group of recipients or when multiple disposition notifications are expected to arrive for the same standalone message, the originating user may receive aggregated disposition notifications based on Service Provider policies. Aggregating disposition notifications may be performed by the originating Participating Function or the Controlling Function.

In the case of delivering an RCS Standalone Message to multiple devices of the same contact/user, the terminating Participating Function shall, for each disposition notification type (i.e. delivery and display notifications), forward the first disposition notification received to the originator of the message and shall suppress the forwarding of subsequent disposition notifications received from the other devices that the message was delivered to.

3.2.1.3 Support for multiple user devices

The RCS standalone messaging supports users with multiple devices. The RCS standalone messaging service shall be available on all of the RCS capable devices/clients of a user. More specifically, an incoming message shall be delivered to all clients of a user, which are online and capable of handling the RCS standalone messaging service. If all clients of a user were offline when a message has to be delivered, the message will be delivered to the first client that comes online if the message has not expired in the meantime. The procedures for handling the multiple devices are described in [RCS5-CPM-CONVFUNC-ENDORS].

3.2.1.4 Deferred Messaging

As opposed to immediate message delivery, the RCS “deferred messaging” is to temporarily hold the message in the terminating Participating Function and deliver it at a later time. Furthermore, the deferred messaging is to defer the delivery of standalone messages when none of the terminating RCS user’s devices is registered and available to receive the messages. In this case, the undelivered messages stay in the RCS Participating Function until they are either delivered to the user devices, are deleted or expire. The procedures for handling the deferred standalone messages are described in [RCS5-CPM-CONVFUNC-ENDORS].

3.2.1.5 Common Message Store

In RCS, a Common Message Store is used to store messages (standalone text or multimedia and chat messages). An RCS user will have control over the messages to be stored in their Message Store. The Common Message Store allows a user to improve their organization of their stored messages. In addition to this, the Common Message Store is used to provide storage for all messages sent and received by a client supporting the RCS text and multimedia messaging service which also includes any other messages that they receive.

The RCS Common Message Store supports synchronization of stored objects with the local storage in all registered RCS devices

The storage is always subject to operator-controlled message size and storage quota limitations.

Relevant storage usage information can be collected to allow a service provider to apply usage based charges.

3.2.1.6 Interworking with legacy messaging services

The purpose of this feature on interworking between the RCS standalone messaging and the legacy messaging services, e.g., SMS, MMS, is to communicate, in a seamless manner, with devices or networks that support legacy SMS and/or MMS messaging services.

3.2.1.7 Personal Network Blacklists handling

NOTE: In the present section, it is assumed that the BPEF as described in section 2.15.1 is provided by the Messaging Server.

When supported, the user defined Personal Network Blacklists are applied by the Messaging Server at both origination and termination of standalone messages.

If any of the recipients of a Standalone message are found in the corresponding Standalone blacklists at either origination and/or termination, the Messaging Server:

- At the originating side:
  - removes the recipient at origination, and continues with processing the message to the remaining recipient(s). If the recipient is the only recipient of the message, then the message is discarded and an error is returned to the originator user.
- On the terminating side:
  - stores the message(s) blocking event in the Blocked Folder.
  - checks whether a notification should be sent to the user about the blocked message or not, based on Service Provider policies.

3.2.2 Interaction with other RCS features

There are no interactions between the RCS Standalone Messaging service and other RCS services.

3.2.3 High Level Requirements

This section contains Standalone Messaging service's high level requirements. These requirements are listed in two separate support aspects for client and server as follows:

3.2.3.1 Client/device support

3-2-1 Common Message Store capability: The ability for RCS users to store and manage their messages if the Common Message Store is deployed and the user has a subscription to the Common Message Store.

3-2-2 Delivery and Display Notifications: Supporting RCS user to request and receive notifications on the disposition state of a standalone message they have sent. Furthermore, the client device should allow both the sending and receiving users to optionally enable/disable the display notifications request and response, respectively.

3.2.3.2 Server support

3-2-3 Number of recipients: For the Standalone Messaging to support both 1-to-1 and 1-to-many (group) messaging features including "reply-to-all" for the group messaging.

3-2-4 Multiple clients/devices: The ability to support RCS users employing multiple RCS capable devices/clients.

3-2-5 Interworking with legacy SMS and/or MMS: The ability to interwork and communicate with other messaging servers supporting legacy SMS and/or MMS messaging services.

3-2-6 Deferred messaging: To defer the delivery of an RCS Standalone Message when none of the terminating RCS devices is registered and available to receive the RCS Standalone Message.

3-2-7 Common Message Store capability: For storing a user's messages and synchronizing them across RCS user's multiple devices.

3-2-8 Delivery and Display Notifications: The server shall ensure that requests for disposition notifications and the notifications themselves are delivered correctly

3.2.4 Technical Realization

3.2.4.1 Standalone messaging

The technical realization of the RCS standalone messaging is based on the OMA CPM Pager Mode and Large Message Mode mechanisms as described in [RCS5-CPM-CONVFUNC-ENDORS]. These messaging modes in conjunction with the 3GPP IMS functional entities as the infra-structure for the messaging functional entities are used as the platform for providing an end-to-end standalone messaging service.

Both CPM Pager Mode and Large Message Mode Standalone Messaging mechanisms are based on the use of the IETF SIP protocol. The Pager Mode messaging uses the SIP MESSAGE method, which imposes a limitation for the maximum message size, while the Large Message Mode messaging uses dedicated SIP/MSRP sessions set up for the delivery of large messages without limiting the message size.

The maximum size of an RCS Standalone Message to be sent using the Pager Mode messaging cannot exceed 1300 bytes. Messages with size exceeding this threshold will be handled by the Large Message Mode messaging. Therefore, an RCS Standalone Message will be sent and delivered using either the Pager Mode or the Large Message Mode depending on the size of the message. This procedure is transparent to the user, i.e., the user does not make the decision to use either Pager Mode or Large Message Mode messaging nor do they see a difference in the service behaviour.

From the user access perspective, the same technology is used for simultaneous delivery to mobile and broadband access clients.

3.2.4.1.1 Pager Mode Messaging

Figure 41 presents an architectural view of the RCS standalone messaging employing Pager Mode messaging.

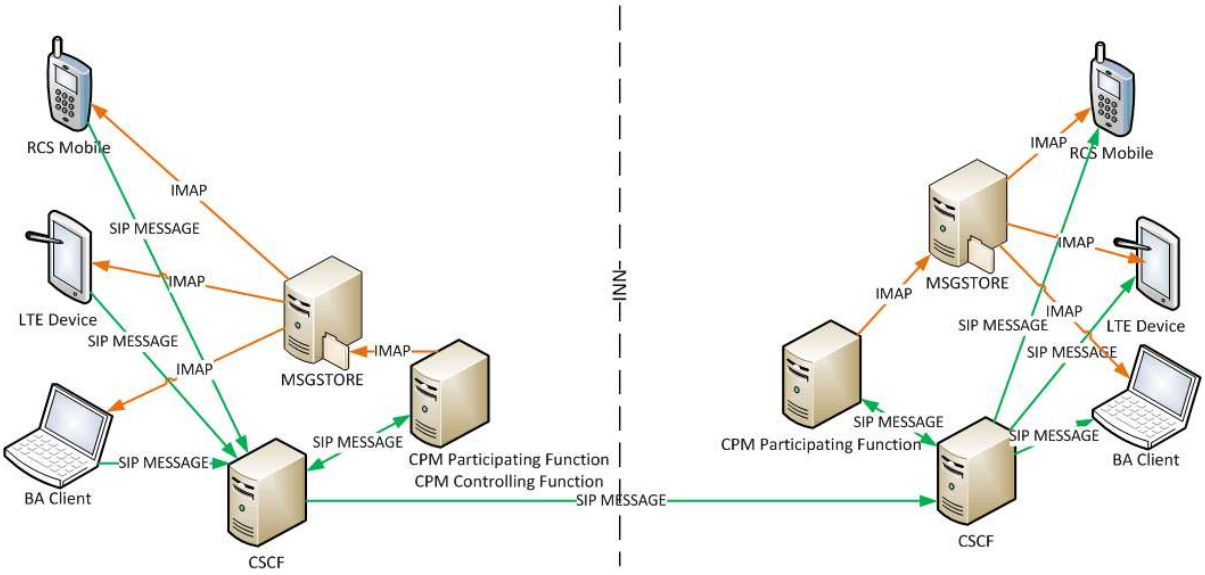


Figure 41: Standalone Messaging using Pager Mode

The detailed procedures for the sending and delivering of a message to the recipient are described in [RCS5-CPM-CONVFUNC-ENDORS]. From the sending user client/device, the message will pass through the Participating Functions at the originating and terminating sides to be delivered to the intended receiving client(s).

If the message is targeted for a group of recipient users, it will be sent from the Participating Function in the originating side to a Controlling Function, also in the originating side that will then perform the procedures for distributing the message to the Participating Functions attending the intended recipient clients.

The RCS Standalone Message delivery and display notifications will follow the reverse path that was used for sending the message.

As described in [RCS5-CPM-CONVFUNC-ENDORS], if the Common Message Store is provided any standalone message that is sent or received will be stored in the corresponding RCS user's Message Store as described in Section 3.2.4.7.

3.2.4.1.2 Large Message Mode Messaging

Figure 42 presents an architectural view of the RCS Standalone Messaging employing the Large Message Mode messaging.

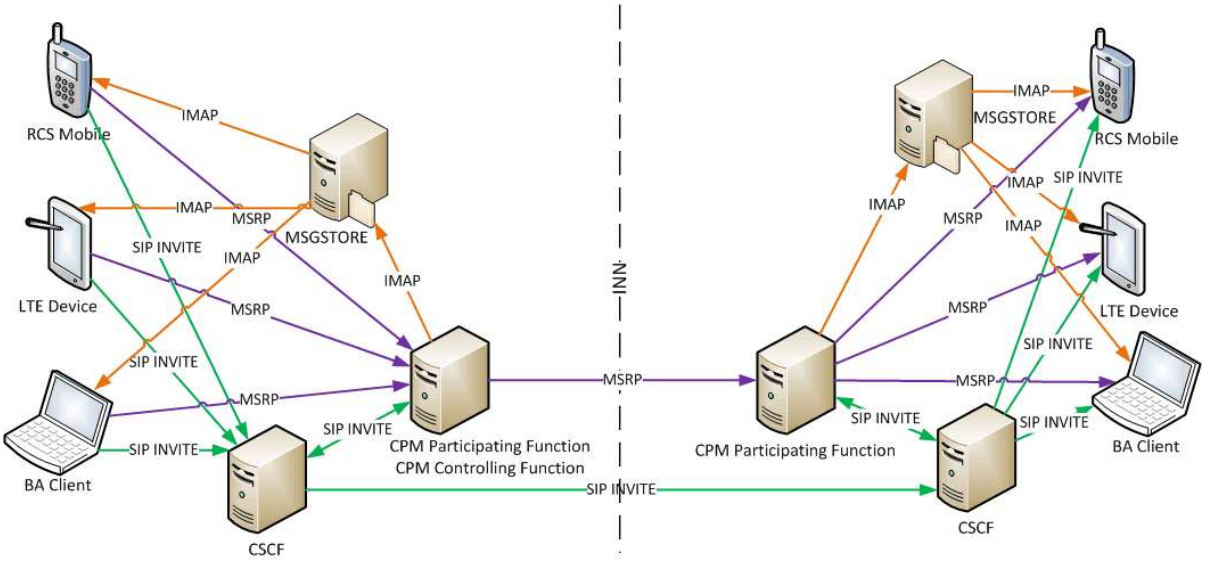


Figure 42: Standalone messaging using Large Message Mode

A large text or multimedia message is sent from an RCS client and delivered to the target client using the Large Message Mode messaging mechanism as described in [RCS5-CPM-CONVFUNC-ENDORS]. Through an MSRP session established following a successful SIP INVITE, the message will be passed through the Participating Functions in the originating and terminating sides to reach the intended recipient. The SIP INVITE includes the size of the standalone message and the content type(s) used in the message.

The terminating Participating Function, amongst other procedures, performs the procedures for deferring messages if none of the intended recipient's RCS capable devices is online.

If the message is targeted for a group of recipient users, it will pass through the Participating Function in the originating side to the Controlling Function also at the originating side before reaching the Participating Function(s) at the terminating side(s). The Controlling Function handles the distribution of the message to various target recipients. As in this case, a list of recipients will be provided along with the delivered message, each recipient has the possibility to send a reply to the sender as well as to all the other users that were addressed in the original message.

The delivery and display notifications of a sent standalone message will follow the reverse path of the sent message.

As described in [RCS5-CPM-CONVFUNC-ENDORS], if the Common Message Store is provided any standalone message that is sent or received will be stored in the

corresponding RCS user's Network-based Common Message Store as described in Section 3.2.4.7.

3.2.4.1.3 Standalone Messaging Service identification

The RCS client shall populate the P-Preferred-Service header field in all CPM requests with the CPM Feature tag defined for the service, as described in [RCS5-CPM-CONVFUNC-ENDORS]. The S-CSCF or AS that performs the service assertion in the originating network shall add the P-Asserted-Service header field set to the value of the asserted CPM service ICSI (i.e. standalone messaging, such as: "*urn:urn-7:3gpp-service.ims.icsi.oma.cpm.msg*" for Pager Mode, or "*urn:urn-7:3gpp-service.ims.icsi.oma.cpm.largemsg*" for Large Message Mode, or "*urn:urn-7:3gpp-service.ims.icsi.oma.cpm.deferred*" for deferred delivery) and remove the P-Preferred-Service header field before further routing the request.

A receiving network element and RCS client should ignore any SIP header fields that they do not understand (e.g. P-Preferred-Service, or P-Asserted-Service header fields).

3.2.4.2 Delivery and Display Notifications

The disposition status notifications for a sent standalone message will follow the reverse path of the sent message. The disposition notifications for the standalone messaging could be used for the 1-to-1 or 1-to-many messaging and for two types of notifications, delivery and display, as specified in [RCS5-CPM-CONVFUNC-ENDORS].

For network optimization purposes, the aggregation of IMDNs as specified in [RFC5438] may be supported for network initiated IMDNs:

- Within the Service Provider's own network, the aggregation of IMDN may be supported (per local policy).
- For inter-Service Provider interoperability, the individual IMDN will always be sent to the target network, where the aggregation of IMDN is up to the target network (per local policy). That is, if the aggregated IMDNs received by the Messaging Server contain IMDNs that need to be sent to another network, the Messaging Server will repackage the aggregated IMDNs accordingly before sending them to the Chat message sender on the other network.
- If the aggregated IMDNs received by the Messaging Server contain both in-network and inter-Service Provider Chat message senders, the Messaging Server will repackage the aggregated IMDNs according to in-network Chat message senders and inter-Service Provider Chat Message senders.

When delivery and display notifications are requested in a Standalone Pager Mode Message (i.e. via SIP MESSAGE), the device identifier shall be set by the sender in the CPIM *From* header, using either a public gruu or a sip.instance value as defined in section 2.11.3.

3.2.4.3 Deferred Messaging

The terminating Participating Function, amongst other procedures, performs the procedure for deferring messages if none of the RCS capable devices of the recipient is online.

When no RCS target recipient client is registered, the terminating Participating Function holding the message for delivery may decide to defer the standalone message for delivery at a later time. For the delivery of a deferred standalone message, the Participating Function has the following options as specified in [RCS5-CPM-CONVFUNC-ENDORS]:

1. To send a notification to the RCS clients of the target recipient and wait for these client(s) to take action,
2. To push the deferred standalone messages once one of the clients of the target recipient RCS user becomes available.

   NOTE: Service provider's policies may guide which option to adopt.

If a deferred standalone message expires before it is delivered, the terminating Participating Function shall handle the deferred message by discarding it.

3.2.4.4 Personal Network Blacklists handling

NOTE: In the present section, the BPEF as described in section 2.15.1 may be provided by the Messaging Server.

When supported, the PNBs are applied by the BPEF at both origination and termination of standalone messages.

The following resource-lists from Shared XDMS (see section 2.14.1) are checked by the BPEF by comparing the URI values used in the request and in the list:

- at Standalone message origination:
  a) the BPEF checks the '*rcs_pnb_outstandalone_blockedusers*' list to verify that the recipient(s) is/are not among the blocked users for this request by comparing URIs contained in the list with the URI value of the Request URI of the SIP request for a 1-to-1 message or with the URIs in the recipient-list body for 1-to-many message.
  b) If true, the BPEF:
    - removes the recipient from the list of recipients before continuing to process and sending out the Standalone message;
    - if the recipient is the only one in the message, then the message is discarded and a *403 Forbidden* response with a warning header set to "122 Function not allowed" is returned to the user.
    - if there are multiple recipients of the message, the number of acceptable recipients is checked by the Messaging Server after the Personal Network Blacklists verification.
- On termination, the BPEF checks the '*rcs_pnb_standalone_blockedusers*' list, to verify if the sender of the Standalone message is among the blacklisted users by comparing the URIs contained in the list with the URI values of the *P-Asserted-Identity* header field of the SIP request.
  a) If true, the BPEF:
    - shall return a *403 Forbidden* response with a warning header set to "122 Function not allowed" and,
    - it suppresses any further IM notifications ("delivered" and/or "displayed") for the blocked messages, and
    - the BPEF stores received blocking event in the dedicated Blocked Folder.

3.2.4.5 Multidevice handling

The RCS supports delivering of standalone messages to multiple devices. As described in [RCS5-CPM-CONVFUNC-ENDORS], the delivery of RCS 5.1 Standalone messages will be done to all the user's RCS devices that are online. Also, when applicable, the message is delivered to a single non-RCS device of the user through interworking with either SMS or MMS as explained in Section 3.2.4.6.

The support of the RCS multidevice environment includes the following major features:

1. When a user sends a message from one of their devices capable of handling the RCS standalone messaging and a Common Message Store is available, all other online devices capable of handling the RCS standalone messaging services shall display the message along with related information such as message state and its disposition.
2. If a Common Message Store is available, all offline clients supporting the RCS standalone messaging service will be capable of showing the messages that the user has sent and received (except for already deleted messages) when the clients are back online.
3. Handling of delivery and display notifications when multiple clients receive a message, the terminating RCS Participating Function shall support forwarding both delivery and display notifications to the originating client, by forwarding the first disposition notification received from one of the devices that the standalone message was delivered to. It suppresses forwarding subsequent disposition notifications received from the other devices to which the message was delivered.

All procedures for sending and receiving standalone messages and their disposition notifications in an RCS multidevice environment, where the RCS user employs multiple devices, are performed as described in [CPM-SYS_DESC] and specified in [RCS5-CPM-CONVFUNC-ENDORS]

3.2.4.6 Interworking with Legacy Messaging services

The [RCS5-CPM-IW-ENDORS] document describes general interworking procedures applicable to both SMS and MMS and the realization details for the SMS and MMS interworking. The interworking procedures for the SMS include references to 3GPP's IP-SM-GW (IP Short Message Gateway) as described in [RCS5-3GPP-SMSIW-ENDORS].

3.2.4.6.1 Interworking procedure

The procedures for the RCS standalone messaging service feature interworking to SMS and MMS legacy messaging services are performed by two interworking functional entities, the Interworking Selection Function (ISF) and the Interworking Function (IWF). After the Participating Function has decided that the message has to be interworked the selection of whether to interwork to SMS or MMS is done in the ISF as described in [RCS5-CPM-IW-ENDORS]. The actual interworking procedure is performed by the SMS and MMS gateways described in [RCS5-3GPP-SMSIW-ENDORS] and [RCS5-CPM-IW-ENDORS]. These functions also interwork the delivery notifications received from the SMS and the delivery and display notifications received from the MMS message recipient(s) and forward them to the sending Participating Function to be passed on to the sending RCS client.

The interworking functions also interwork any incoming SMS or MMS messages to RCS messaging.

3.2.4.6.2 Interworking with SMS

When the target recipient device for an RCS Standalone Message is a non-RCS capable, an SMS capable device, the process of interworking with legacy SMS is invoked according to [RCS5-CPM-CONVFUNC-ENDORS]. In Figure 43, an architectural view of the RCS standalone messaging service interworking with the legacy SMS is shown. The legacy mobile device is shown as a non-RCS device.

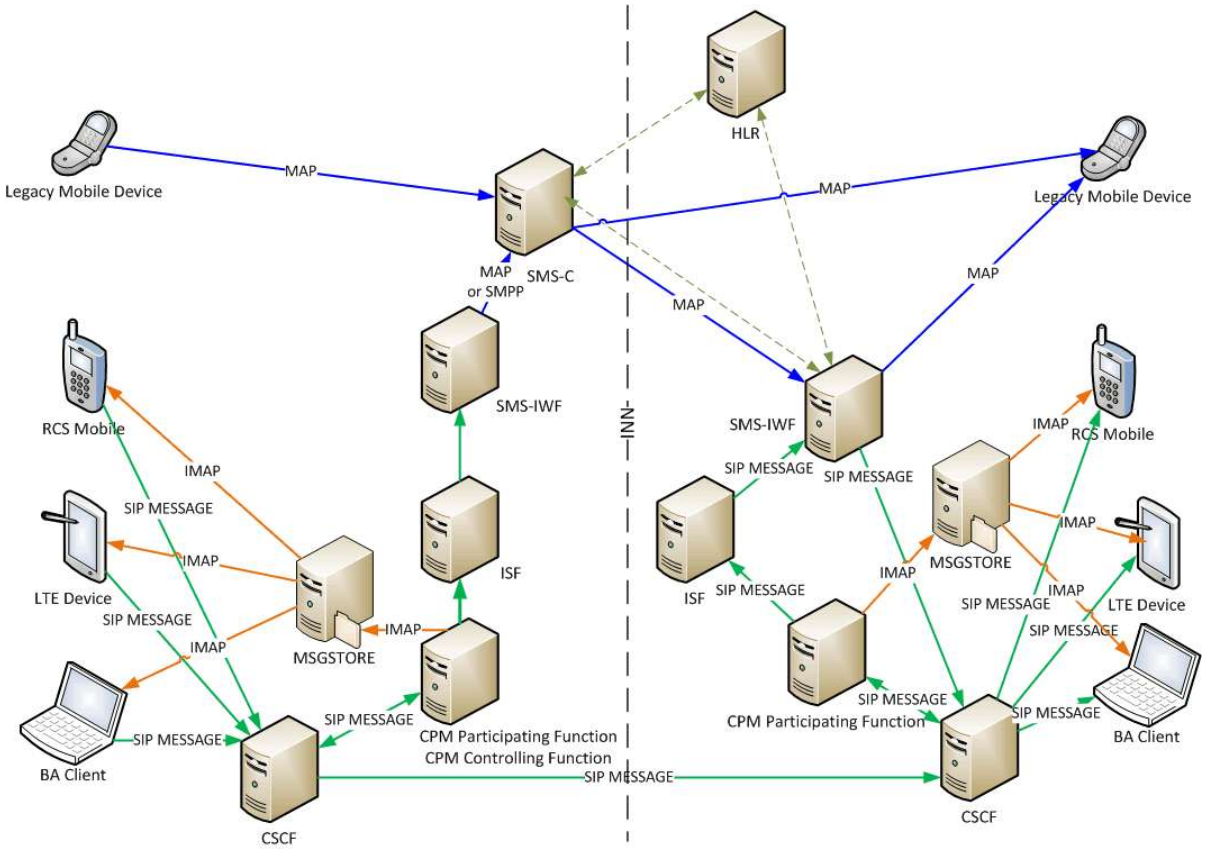


Figure 43: Standalone Messaging interworking with SMS

When the SMS interworking function (IP-SM-GW or SMS-IWF) receives a SIP MESSAGE request with the OMA CPM ICSI "*3gpp-service.ims.icsi.oma.cpm.msg*", it checks the size of the received payload of the SIP MESSAGE request. If the size of the payload is too large to be sent as one SMS message, the payload will be divided into concatenated SMS messages. The SMS-IWF will send the request(s) generated based on the received SIP MESSAGE request towards the SMS-C (Short Message Service Centre) using either the SMPP (Short Message Peer-to-Peer) or MAP (Mobile Application Part) protocols, depending on the type of SMS network in which it is deployed, as specified in [RCS5-CPM-IW-ENDORS] or [RCS5-3GPP-SMSIW-ENDORS] respectively.

NOTE: For clarity, Figure 43 mainly shows the latter deployment option since the differences between both options are in the existing SMS deployments and therefore have no impact on the Standalone Messaging service.

Breakout to SMS can be done at the originating side if the addressed user is not an IMS user. This is determined based on the standalone messaging capability information, on local information the Messaging Server may have about the recipient, or when the Messaging Server receives an error response. Otherwise, the breakout at the terminating side is done, if either the addressed user is an RCS user using SMS instead of RCS standalone messaging service or the user is using a mixture of legacy and RCS devices.

The following error responses to the SIP MESSAGE (or, for the IP-SM-GW realisation, optionally for a Large Message Mode message the SIP INVITE) request indicate that the recipient is not an RCS contact and these responses can be used to trigger interworking:

- 404 Not Found;
- 405 Method Not Allowed;
- 410 Gone;

- 414 Request URI Too Long;
- 415 Unsupported Media Type;
- 416 Unsupported URI Scheme;
- 488 Not Acceptable Here;
- 606 Not Acceptable.

The case for delivering text messages to a (primary) broadband client of a non-Standalone Messaging user is beyond the SMS interworking gateway of the standalone messaging and its platform. It is not shown in Figure 43 to avoid overloading it. In that scenario the MAP (Mobile Application Part) or SMPP request from the SMS-IWF to the legacy Mobile Device for the incoming SMS message would be replaced by a SMSoIP (SMS over IP) request, which is relayed to the legacy BA Client via the Serving Call Session Control Function (S-CSCF).

3.2.4.6.3 Interworking with MMS

When the target recipient device for standalone messaging is not an RCS device and the message to be sent is a multimedia message, the process of interworking with legacy MMS is invoked according to [RCS5-CPM-CONVFUNC-ENDORS].

Figure 44 presents an architecture view of the interworking for multimedia messaging. As shown, the legacy mobile device at the terminating side may either be an RCS user's primary device that uses MMS instead of RCS Standalone Messaging or a non-RCS device capable of receiving MMS.

Depending on the size of the standalone message, it could be either a text message with a large payload or a multi-media standalone message. In the former case the interworking with SMS would apply as described in section 3.2.4.6.2 if the message were small enough for a concatenated SMS. Otherwise, the interworking would be to the MMS service, hence sending a SIP INVITE request to the RCS MMS-IWF.

When the RCS MMS-IWF receives a SIP INVITE request containing the OMA CPM ICSI "*3gpp-service.ims.icsi.oma.cpm.largemsg*" for a Large Message Mode standalone message, it will send a 200 "OK" response if no errors are found in the SIP INVITE request or an appropriate error response. This is followed by the MMS-IWF's subsequent receiving of an MSRP SEND request for the establishment of the MSRP session, and the process then continues as described in [RCS5-CPM-IW-ENDORS].

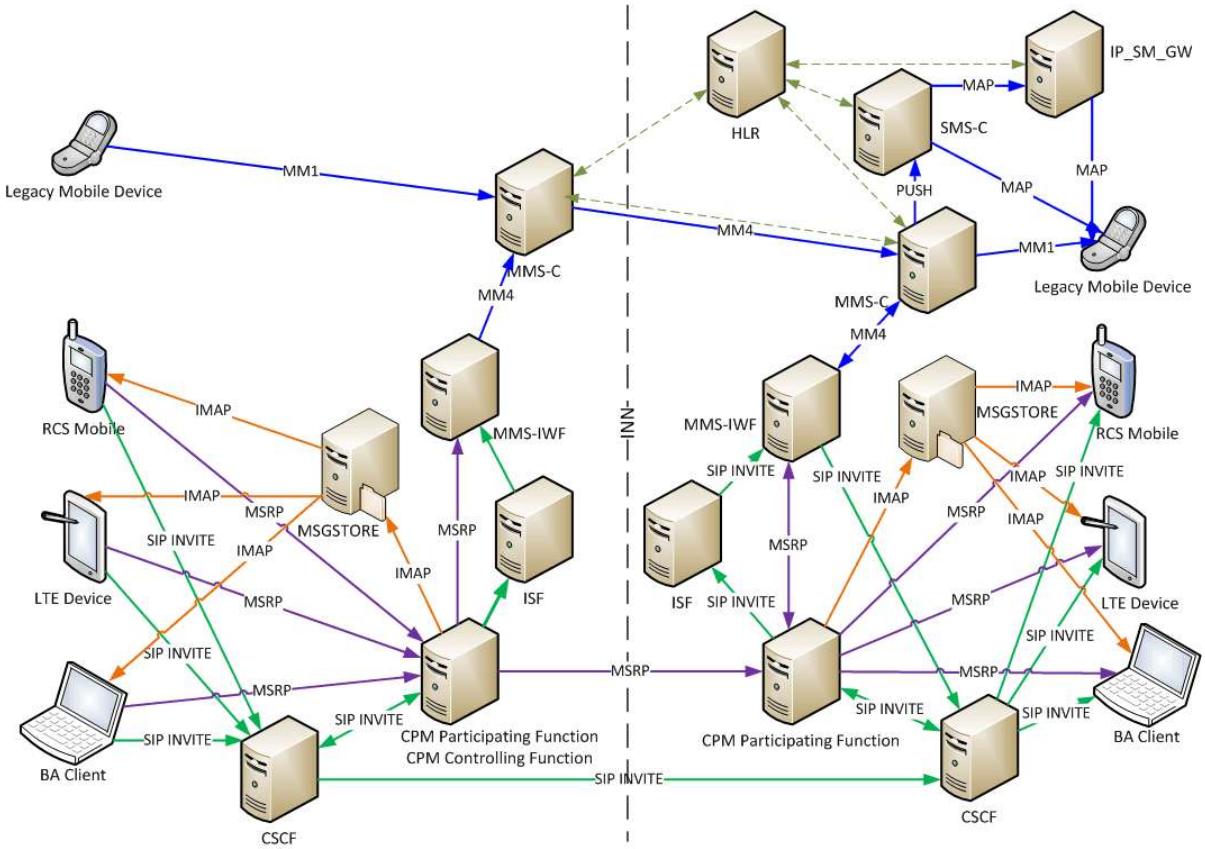


Figure 44: Standalone messaging interworking with MMS

Breakout to MMS can be done at the originating side if the addressee is not an IMS user either based on local information the Messaging Server may have about the recipient, or when it receives an error response. Otherwise, the breakout at the terminating side is done if either the addressee is an RCS user using MMS instead of RCS standalone messaging service or the user is using a mixture of legacy and RCS devices.

The following error responses to the INVITE request indicate the recipient is not an RCS contact and can be used to trigger interworking:

- 404 Not Found;
- 405 Method Not Allowed;
- 410 Gone;
- 414 Request URI Too Long;
- 415 Unsupported Media Type;
- 416 Unsupported URI Scheme;
- 488 Not Acceptable Here;
- 606 Not Acceptable.

Similar to SMS interworking, in the MMS interworking of Figure 44, the case for delivery of the multimedia message to a Broadband Access client of a non-RCS user is not shown to not overload the figure.

3.2.4.7 Common Message Store

RCS supports a “Common Message Store” as described in Section 5.5 of [CPM-SYS_DESC] and specified in [RCS5-CPM-MSGSTOR-ENDORS]. Using an IMAP connection, an RCS user can access and manage their stored objects, as described in [RCS5-CPM-MSGSTOR-ENDORS] regardless of their RCS service registration.

Regarding the client synchronization mechanism that applies, client synchronization guidelines are described in 3.2.6.2.3.

A receiving primary device sending or receiving messages via the SMS or MMS (e.g. in case of no data connection) may, subject to Service Provider policy regarding automatic SMS or MMS storage in the CMS, also receive these messages via the synchronization from the Common Message Store. Since legacy messages do not contain Conversation-ID and Contribution-ID a different mechanism is required to link together the two representations of the same message.

The following sections describe the mechanism used for a device to correlate legacy SMS/MMS messages with the same messages already stored in the Common Message Store.

3.2.4.7.1 Common Message Store and pager/multimedia-messages

To identify the messages in the Common Message Store that will match legacy SMS/MMS messages sent or received by the device via legacy means, it shall be possible to keep information about the submission or delivery path (SMS or MMS) for converted messages in the Common Message Store.

The information shall be stored for messages by means of the message context for internet mail (see [RFC3458]).

In RCS the following values of the "message-context" are applicable:

- For received messages:
  - pager-message: the message is delivered to a primary device via SMS
  - multimedia-message: the message is delivered to a primary device via MMS

  NOTE: In RCS the message context is only used in the relation between the terminating CPM Participating Function, the Common Message Store and the recipient user's device. It does not provide information of the message context on the originating side nor on the NNI.

- For sent messages:
  - pager-message: the message was sent via SMS
  - multimedia-message: the message was sent via MMS

3.2.4.7.1.1 Client initiated storage of SMS/MMS

The RCS Client only stores messages in the user folders of the Common Message Store in a RCS specific folder named: ‘RCSMessageStore’, which cannot be removed by RCS Clients.

NOTE: This procedure is applied when synchronizing as described in section 3.2.6.2.3.

If the SMS MESSAGE STORE or MMS MESSAGE STORE configuration parameter defined in Table 85 is set to “always store in the Common Messaging Store" (i.e. 2), or if the SMS MESSAGE STORE or MMS MESSAGE STORE configuration parameter is set to “store if not found in the Common Message Store” (i.e. 1), and a client determines that a locally stored message (SMS or MMS respectively) is not already stored in the Common Message Store, the client stores it in the Common Message Store under the RCSMessageStore folder identified by the identity of the Contact in the Conversation. If this folder is not yet created in the RCSMessageStore, the Client shall first create it.

If the SMS MESSAGE STORE or MMS MESSAGE STORE configuration parameter is set to "2" then the client does not need to apply the message correlation as described in section 3.2.4.7.2 for SMS and MMS respectively.

For the case that the client stores a message that is sent or received as an SMS/MMS and there is no copy of that message in the Common Message Store (SMS MESSAGE STORE or MMS MESSAGE STORE configuration parameters as described in Table 85 equals to 1 or 2), the client shall store the message as described above while setting the message-context parameter to pager-message/multimedia-message as described in section 3.2.4.7.1.

3.2.4.7.2 Correlating SMS/MMS messages with messages stored in the Common Message Store

The following mechanisms describe how to correlate messages received via legacy means with messages stored in the Common Message Store.

For SMS messages:

- The entity in the network storing the message shall store the prefix of the SMS text body (up to 140 bytes) in the RCS-SMS-Content header of the message (see section C.4.1).
- The device shall use this RCS-SMS-Content header value, along with To/From headers to find the corresponding locally stored SMS message
- The algorithm is as described in section 3.2.4.7.3.

For MMS messages:

- Each MMS message is delivered with a unique MMS Message ID (message-id)
- The Common Message Store shall store the MMS Message ID in the message-id field defined in [RFC2822]. The field shall encode as defined in [RFC2822], i.e. the Message-ID-value defined in [MMSENC] bracketed by “<” and “>”.
- The device uses the Message ID from each MMS to find the Unique Identifier (UID) of each corresponding MMS in the Common Message Store by matching it with the Message-Id header with each stored MMS.

Since the Common Message Store remains the master storage for these legacy messages, it is up to the client implementation whether or not to discard matched messages received via legacy means.

While correlation collisions will generally be infrequent, there are particular circumstances where they are quite likely to occur. Therefore, in addition to this basic process, additional logic is required to handle correlation collisions for SMSs, see section 3.2.4.7.4.

3.2.4.7.3 Correlation Algorithm for SMS

The correlation is based on the following field values:

- To: It should be the format as taken from the address field defined in [3GPP TS 23.040]. If TON (Type Of Number) indicates “international”, then a “+” is inserted before the number string. If TON indicates “unknown” only the number string is used. If the address is “alphanumeric”, then the address shall be encoded to UTF-8 format
- From: It should be the format as taken from the address field defined in [3GPP TS 23.040]. If TON indicates “international”, then a “+” is inserted before the number string. If TON indicates “unknown” only the number string is used. If the address is “alphanumeric”, then the address shall be encoded to UTF-8 format.
- The RCS-SMS-Content header value which is generated from the Text Payload contained in the user data of the short message with up to 160 characters as defined

below. Characters or data contained in SMS user data information elements (i.e. SMS and EMS control data as well as EMS content data) are not considered for the correlation algorithm.
Entities storing the message and clients correlating messages shall compose the RCS-SMS-Content header value as follows:

- For messages with no text payload in the SMS user data an RCS-SMS-Content header with no value shall be generated.
- The text payload of the user data of the short message is converted from its original encoding (GSM 7 bit default alphabet or UCS2, see [3GPP TS 23.038]) into UTF-8 format.
- A UCS2 (2-byte Universal Character Set) "Null" field (0x0000) is represented by UTF-8 "space" (0x20).
- UCS2 and GSM 7 bit default alphabet characters "CR" and "LF" and the sequence "CR LF" are each represented by one UTF-8 "space" (0x20)
- National single and locking shift tables indicated in the user data header Info Element are not used for the generation of the RCS-SMS-Content header value. Instead the GSM 7 bit default Alphabet and the GSM 7 bit alphabet extension table are used.
- If a <shift> <character> sequence points to an unassigned code point in the GSM 7 bit alphabet extension table then both the shift codes and the unassigned character code shall be represented as one UTF-8 "space" (0x20) for the purpose of the RCS-SMS-Content header generation.
- In case of concatenated SMS messages, only the first 160 characters of the message after conversion in UTF-8 (regardless of their position in the concatenated short message) will be used to generate the RCS-SMS-Content header value.
- If the resulting string contains only US-ASCII characters (0x20 – 0x7e) it will be taken as the value of the RCS-SMS-Content header.
- If the resulting string contains at least one non US-ASCII character, the RCS-SMS-Content header value shall be encoded as defined in [RFC2047]. The value shall be encoded by the use of the UTF-8 character set (charset = utf-8) and base64 encoding (encoding = b). In this case the client should use for correlation of messages the "encoded-text" part of the header value. For details of the RCS-SMS-Content header encoding refer to section C.4.1.
- Examples of RCS-SMS-Content headers:
  the RCS-SMS-Content header of a short message with the text payload:
  `To your health, my friend`
  will encoded as follows
  `Rcs-Sms-Content: To your health, my friend`
  the RCS-SMS-Content header of a short message with the text payload
  `На здоровье, мой друг`
  will be encoded as follows
  `Rcs-Sms-Content: =?utf-8?b?`
  `0J3QsCDQt9C00L7RgNC+0LLRjNC1LCDQvNC+0Lkg0LTRgNGD0LM=?=`

Additional considerations:

- For the correlation of outgoing messages the From field is not used
- For the correlation of incoming messages the To field is not use
- The correlation is achieved by comparing the relevant elements (To/From, and RCS-SMS-Content header value), using a case-sensitive comparison.

The matching algorithm should take into account differences in the presentation of the address string according to different types of numbers.
The creation of a RCS-SMS-Content header value used for the correlation via a full string match requires in some scenarios access to the native SMS Transfer Protocol Data Units (TPDU, i.e. the TP-UD, TP-DCS data units). Client implementations that do not have access to the TPDU but only to the "interpreted" payload of the short message may compensate this by alternative matching algorithms, e.g. a pattern match algorithm based on a smaller sequence of characters taken from the message content. Such a search algorithm increases the probability of collisions or matching errors, but may be compensated by multiple probes with different sequences of characters taken from the message content with a statistical evaluation.

3.2.4.7.4 Dealing with Collisions

The correlation field values are used to correlate between SMS messages on the Common Message Store and on the device. Specifically, when the device synchronizes with the Message Store Server it will obtain UIDs and the correlation field values for those SMS messages that are new or have changed since the last synchronization. The device will then attempt to correlate the UIDs and correlation field values with any messages it has received or subsequently receives from the network. Therefore, if any of the messages have the same correlation field values (this is considered a correlation “collision”) then the device cannot distinguish between them when matching to its local messages.

The device should compare the direction (originating or terminating) in addition to comparing the correlation field values, meaning that correlation collisions can only occur on messages with the same direction.

Correlation collisions can occur in these two cases:

1. Messages in the same thread with the same content, typically when they are chronologically close (so returned on the same synchronization) SMS messages in the same thread with the same content, such as successive replies both saying “OK”.
2. Messages in the same thread with content that is different only after the first 140 bytes. This is more likely when higher numbers of messages are being compared, for example, a likely worst case example would be when a phone has been switched off for a long period (e.g. a vacation, a repair). This rare scenario is not addressed here further.

If there are collisions, the device should identify the chronologically first received message on the device with the lower UID on Message Store Server.

For example, suppose Message Store Server returns two new messages both with the same value C for the RCS-SMS-Content header but with UIDs x and y, $x < y$, and the device has received two messages with the same value C for the RCS-SMS-Content header at times t1 and t2, $t1 < t2$. Then the device should identify $t1 = x$ and $t2 = y$.

The same principle applies when the number of correlation collisions on the device is different from the number on the Message Store Server; those are usually cases of temporary lack of synchronization between the device and the Common Message Store.

As an example, suppose as above the Common Message Store has the same two new messages but the device has only received one message with value C for the RCS-SMS-Content header. It should identify that with UID x, in the presumption that the network will shortly deliver a second message with value C for the RCS-SMS-Content header which it will then identify with UID y. Similarly, if the Common Message Store only has UID x producing value C for the RCS-SMS-Content header but the device has both t1 and t2, the device should identify t1 with message x and expect a subsequent synchronization to return message y which it will then identify with t2.

Note that some legacy messages might not have been stored in Common Message Store by the network. Therefore the length of time between the messages should be considered by the client when determining whether the messages are duplicates. Note also that the device would have to take into account messages the device might have that it received before the Common Message Store was in place.

The impact of correlation collisions in this method may result in a wrong correlation; in the case above, to identify t1 = x and t2 = y when the correct mapping was in fact t1 = y and t2 = x. In this case, the view from one device and another will be out of sync: a user making a state change to t1 on one device will see it applied to t2 on the other device, when they would expect it to apply to message y.

For example, take the case of successive identical messages. If the user marks on one device the earlier of these messages as a favourite, then the device view might be as follows:

"are you still on for tonight?"

"yes" <- FAVOURITE

"do you have the tickets?"

"yes"

whereas on another device the view would be:

"are you still on for tonight?"

"yes"

"do you have the tickets?"

"yes" <- FAVOURITE

No messages are lost, so there is no need to define any more advanced methods.

3.2.5 NNI and IOT considerations

For the Standalone Messaging service three NNI interfaces are possible:

- The NNI described in section 2.12 carrying the standalone messaging service across RCS compliant networks
- SMS NNI
- MMS NNI

Which of these interfaces is used is decided based on the Service Provider's policies and the applicable interworking agreements.

3.2.6 Implementation guidelines and examples

3.2.6.1 Possible supported entry points to the Standalone Messaging

From the RCS user experience, the following three possible entry points to the Standalone Messaging may be supported:

1. Standalone Messaging screen/window
   There may be a dedicated "Standalone Messaging" application point of entry in the device menu. From this Standalone Messaging screen/window, a standalone message can be initiated or received using the relevant menu items and the device's supported keypad/keyboard. This application may also provide access to the user's message store for viewing and managing stored messages, e.g. message history.
2. Integrated messaging screen/window
   There may be a dedicated integrated messaging application point of entry in the device

menu. From this integrated screen/window, a message can be initiated or received using the relevant menu items and the device's supported keypad/keyboard. This application may also provide access to the user's message store for viewing and managing stored messages, e.g. message history.

3. Address book window
   Using this entry point, a message may be initiated with any contact. The experience when interacting through this entry point is identical to that of the messaging screen/window.

   NOTE: when displaying the messages exchanged, the time indication can be set according to the CPIM *DateTime* header and the *datetime* element in the delivery notifications as described in section 3.3.6.6 allowing correct ordering of the messages even if the device's clock is not set correctly.

3.2.6.2 Clarifications on procedures at Message Store Client

Clients shall comply with the operations and procedures described in sections 5.5.2 of [CPM-SYS_DESC] and 6 of [RCS5-CPM-MSGSTOR-ENDORS]. Some further clarifications on the client expected behaviour when it interacts with the Message Store Server are presented in the following sections.

3.2.6.2.1 Storing new messages (Object Store Operation)

For the cases that the client stores new messages and there is no existing folder where these messages can be stored (e.g. a brand new conversation with user B offline and interworking procedures in place), the client needs to allocate the name to the new folder and follow the naming procedures as described in 6.3.1 of [RCS5-CPM-MSGSTOR-ENDORS].

3.2.6.2.2 Message Archive (Object Copy Operation)

For user initiated archive requests, procedures as described in section 6.4.3 of [RCS5-CPM-MSGSTOR-ENDORS] apply. For the case that permanent message storage is required due to Service Provider policy, the client is not expected to automatically perform any Object Copy operation.

3.2.6.2.3 Synchronization

NOTE: Client synchronization guidelines are built under the assumption that roaming scenarios are out of scope. Synchronization triggers may change under roaming scenarios

There are three possible types of triggers for synchronization. Specifically:

1. Data Connection State Triggered Synchronization
   This type of synchronization trigger includes all the cases where the device moves from disconnected to connected state. This includes the following triggers:
   a) the device is powered on or the client is launched
   b) regain of data connection due to toggle of the operating system's "data on/off switches", e.g. "data traffic switches" or "flight mode"
   c) in the case of other data connection state triggers and if the client detects a loss of data connection when setting up or during a connection with the message store or during other network interactions it should (re-) connect to the message store on connectivity re-gain (see also section 2.4.7.6).

All folders are expected to be synchronized and no folder prioritization for synchronization applies. Connection state triggered synchronization happens in the client background and once triggered, the client shall initiate the login and first time synchronization procedures (see flow in section B.4.2). Once the synchronization procedure is completed, client is expected to logout. Clients with bearer control capabilities should attempt to logout from an existing IMAP session prior to data connectivity loss.
Note that for the cases of short data connectivity loss, clients shall rely on the other synchronization triggers (described below) and consequently this type of synchronization trigger does not apply.

2. Periodic synchronization
Periodic synchronization shall be triggered within a configurable time interval set by the Service Provider (please refer to MESSAGE STORE SYNC TIMER configuration parameter as described in Table 85). As for synchronization due to data connectivity re-gain, all folders are expected to be synchronized and no folder prioritization for synchronization applies. Periodic synchronization happens in the client background and once triggered, the client shall initiate the login and first time synchronization procedures (see flow in section B.4.2). Once the synchronization procedure is completed, the client should logout.
3. Synchronization triggered by User activity
This type of synchronization shall be triggered once the user enters a particular conversation. For this type of synchronization, all folders are considered for the synchronization procedure but prioritization is given to the conversation that the user enters right before synchronization triggering. The synchronization of the rest of the conversations happens right after the synchronization of the prioritized conversation is completed and it takes place in the background.

NOTE: Client synchronization procedures shall consider the folder structure as defined in Object Store Operation described in 6.3.1 of [RCS5-CPM-MSGSTOR-ENDORS].

For the case that SMS MESSAGE STORE parameter (please refer to SMS MESSAGE STORE configuration parameter as described in Table 85) is set to 1, the client shall apply the correlation algorithm for all sent and received SMS messages as described in 3.2.4.7.3 and upon next synchronization correlate them with messages in the network Message store indexed as message-context=pager-message.

For the case that SMS MESSAGE STORE parameter (please refer to SMS MESSAGE STORE configuration parameter as described in Table 85) is set to 2, the client shall not apply the correlation algorithm for any sent or received SMS message as described in 3.2.4.7.3. The client shall store every sent and received SMS and no attempt for correlation shall be made.

3.3 1-to-1 Chat

3.3.1 Feature description

3.3.1.1 General

The Chat service enables users to exchange messages between two users instantly.

The following RCS 1-to-1 Chat features are described:

- Store and forward
This feature requires a Messaging Server to store messages and notifications (delivery and display) when the destination user is not online and deliver them to the user when he comes online again (i.e. store and forward).

- Interworking of Chat to SMS/MMS
  This feature requires a Messaging Server to interwork the messages to and from SMS or MMS.
- Message revocation request of Chat messages
  This feature allows a client or originating Messaging Server to request for an undelivered Chat message to be revoked.
  The revocation process is not user driven but a technical enabler for clients or the Messaging Server on the originating network.
- Message revocation processing of Chat messages
  This feature requires a Messaging Server to process MessageRevoke requests and respond with a MessageRevokeResponse request based on the chat message delivery status.
- "Delivered" message notification
  This allows the sender of a message to be notified when their message has been delivered to the recipient.
- "Displayed" message notification
  This allows the sender of a message to be notified when their message has been displayed on one of the recipient's devices. Note that this notification cannot certify that the recipient has actually read the message. It can only indicate that the message has been displayed on the recipient's terminal User Interface (UI).
- Delivery of notifications (delivered and displayed) outside a session
  It should be possible to deliver notifications independently of whether a 1-to-1 chat is established or not.
- IsComposing indications
  This allows a user in a chat conversation to see when another user is typing a new message/reaction.
- Local Black List
  The terminal/client may support a locally stored Black List to handle incoming chat requests. Users are allowed to qualify undesired incoming chat as spam. This prevents subsequent messages from those originators to be shown or even notified to the user. Also, this undesired traffic will not be acknowledged to have been read. The Black List behaviour applies not only to Chat but also to File Transfer.
- PNB
  The PNBs stored in the network and set by the RCS user contains the lists of URIs for contacts (or lists), that an RCS user has set for blocking purposes. The BPEF uses the PNB lists for chat incoming and outgoing traffic blocking.
- Local Conversation History
  The terminal/client supports a locally stored conversation.
- A Common Message Store
  A Common Message Store for the chat sessions may be used to synchronize the messages between devices. It also allows the user to keep a back-up of important conversations in the network.
  In the device, alignment is expected between the local Conversation History and the synchronization with the Common Message Store.
- User Alias (Display Name)
  A user defined display name may be sent when initiating a communication with another user.
- Flexibility to allow multimedia messages within a chat conversation
  Multimedia message exchange is supported in a chat session. However, whether or not multimedia messages are allowed during a Chat session is up to a Service Provider and

controlled by a configuration parameter (see MULTIMEDIA IN CHAT in Table 85 of Annex A).

3.3.2 Interaction with other RCS features

3.3.2.1 Switching to Group Chat

A Group Chat can only be initiated from a user on a Service Provider which has deployed a Messaging Server. It is optional for a Service Provider to provide the Group Chat functionality. Therefore from the terminal perspective, if the configuration parameter CONF-FCTY-URI (see Table 84) is not configured or configured with a dummy value, the terminal should not allow the user to add additional parties to the chat or start a Group Chat.

A 1-to-1 Chat can be converted into a Group Chat by either of the two Users A and B by adding new users to it. User A and User B are given the option in their UI to add one or more chat partners to the conversation. A user may be limited to the contacts known by their devices to be RCS users. Otherwise the originating user's Messaging Server needs to be prepared to potentially interwork messages to non-RCS Users via SMS or MMS.

A real time check of contacts capabilities may be performed when initiating a Group Chat (section 3.3.6.3). A new Group Chat composing window is created in the initiating device, for example, User A's device, and the result of this check is visible here.

When User A sends the first message a new Group Chat is opened between all the selected users, and User A and User B as described in section 3.3.6.3.

For User B a new Group Chat composing window is created in the user's device.

3.3.2.2 File Transfer within 1-to-1 chat and interaction with the blacklist

NOTE: In the present section, the BPEF as described in section 2.15.1 may be provided by the Messaging Server.

During a 1-to-1 chat, either user is able to initiate a File Transfer from the chat composing window towards the other user. The File Transfer is established using a new SIP session and is carried in a new MSRP session which is different from the one used for the chat session.

If PNB is supported, the handling by the BPEF is same as in section 3.5.4.5. Note that in the case of File Transfer during chat, the sender of the file transfer needs to be checked again but against the FT blacklist this time '*rcs_pnb_ft_blockedusers*'.

On the device involved in the chat, the receiving user receives the File Transfer invitation inside the chat window with the sending user and is able to accept or decline it from that window. In a multidevice environment, the File Transfer invitation is also shown on the other devices of the user allowing them to accept or decline the invitation also from those devices.

If the user accepts the File Transfer, the terminal will either ask the user the location to store the file or use a default directory. Once received, the user can open the file from the chat composing window.

Please note that the spam/blacklist behaviour applies to File Transfer, and not only to Chat messages. If an invitation to receive a file is received from a blacklisted user, the client/device implementation should, from the UI point of view, not notify the user on receipt of a File Transfer invitation from a blacklisted sender. Instead it should log the event in the spam folder (e.g. "User A tried to send a file on TIME/DATE").

See section 3.5 for more details on File Transfer.

3.3.3 High Level Requirements

The following list of high level requirements applies to 1-to-1 chat:

- Clients/devices:

3-3-1 "Delivered" message notification request and response

3-3-2 "Displayed" message notification request and response
Note that the client device should allow the user to enable or disable the displayed notifications request and response

3-3-3 Delivery of notifications (delivered and displayed) outside a session

3-3-4 IsComposing indications

3-3-5 Procedures associated with the store and forward of both messages and notifications performed by the Messaging Server

3-3-6 Sending MessageRevoke requests

- Messaging Server: In addition to the requirements presented above

3-3-7 Store and forward of both messages and notifications
Please note this is a function which is provided on the terminating Service Provider's network however a Messaging Server may additionally provide originating store and forward to avoid dependencies with another Service Provider network's implementations.

3-3-8 Interworking of Chat to SMS/MMS

3-3-9 Sending MessageRevoke requests

3-3-10 Handling of MessageRevoke requests

3.3.4 Technical Realization

Two different technical realizations of 1-to-1 chat are available: OMA SIMPLE IM as described in [RCS5-SIMPLEIM-ENDORS] or OMA CPM as described in [RCS5-CPM-CONVFUNC-ENDORS]. The first sub-section describes the features that are common to both technical realizations, while the following two sub-sections describe what is unique to the individual technical realizations. The CHAT MESSAGING TECHNOLOGY configuration parameter defined in Table 85 determines the technical realization used for 1-to-1 Chat.

3.3.4.1 Technical Realization of 1-to-1 Chat features common to both OMA SIMPLE IM and OMA CPM

At a technical level the Chat service implemented using OMA SIMPLE IM or OMA CPM relies on the following concepts:

- SIP procedures for the setup of sessions using MSRP for the message exchange;
- In the SDP of the SIP INVITE request and response, the *a=accept-types* attribute shall include only *message/cpim* and *application/im-iscomposing+xml*, i.e., "*a=accept-types:message/cpim application/im-iscomposing+xml*".
- If initiating or accepting this chat would have increased the number of concurrent chat sessions above the Service Provider configured maximum limit (see MAX CONCURRENT SESSIONS in Table 85), the device would close one of the other active chat sessions (for example, a chat that has not been used for the longest period of time) before initiating a new one.
- When a session is set up, messages are transported in the MSRP session. Each MSRP SEND request contains a request to receive an Instant Messaging Disposition Notification (IMDN) 'delivery' notification, and possibly a request to receive an IMDN 'display' notification. A client should therefore always include "positive-delivery" in the value for the CPIM/IMDN Disposition-Notification header field. That means that the value of the header field is either "positive-delivery" or "positive-delivery,display" depending on whether display notifications were requested. The value of "negative-delivery" is not used in RCS for 1-to-1 Chat.
  The receiving devices must generate an MSRP SEND request containing the IMDN

status when the user message is delivered and if requested, another MSRP SEND request when the message is displayed.

NOTE: If there is not an already established MSRP session between sender and receiver, the Pager Mode (i.e. SIP MESSAGE) is used to transport IMDNs (delivery notification, display notifications)

- In normal circumstances between 2 users at most only a single session is active at a time. A client shall therefore not initiate a new Chat session towards a user with whom there is already an established Chat session.
- IMDN [RFC5438]: RCS relies on the support of IMDN as defined in [RFC5438] and [RFC5438Errata] to request and forward disposition notifications of all the exchanged messages (See also section C.2 for the errata mentioned in [RFC5438Errata]);
- In MSRP requests, the client shall set both the CPIM From and CPIM To headers to *sip:anonymous@anonymous.invalid* to prevent revealing the user's identity when transmitted over unprotected links. A client receiving a CPIM message in a one-to-one Chat should therefore ignore the identity indicated in the CPIM headers.
- The CPIM/IMDN wrapper shall be UTF-8 encoded to avoid any potential internationalization issues.
- The device identification uses the mechanisms described in section 2.11.3;
- IMDN message identification for all messages (including those conveyed in the SIP INVITE and notifications delivered via SIP MESSAGE) as defined in [RFC5438];
- The originating Messaging Server shall always set the CPIM DateTime header in the chat messages and notifications it receives by overwriting the value provided by the client. A client receiving these requests should therefore rely on these headers rather than on locally available time information.
- Both the Originating and the Terminating function shall ensure that messages are received in correct order by the RCS client regardless if the messages are store and forwarded or not.
    - To achieve this, the terminating side shall wait for the delivery notification or 180 ringing response until a new message is sent if the message is carried in the INVITE.
    - As in the case for messages composed while offline (see section 2.7.1.1) when a message is carried in the INVITE request, the client shall wait for a provisional response before sending a new message
- For network optimization purposes, the aggregation of IMDNs as specified in [RFC5438] may be supported for network initiated IMDNs:
    - Within the Service Provider's own network, the aggregation of IMDN may be supported (per local policy).
    - For inter-Service Provider interoperability, the individual IMDN will always be sent to the target network, where the aggregation of IMDN is up to the target network (per local policy). That is, if the aggregated IMDNs received by the Messaging Server contain IMDNs that need to be sent to another network, the Messaging Server will repackage the aggregated IMDNs accordingly before sending them to the Chat message sender on the other network.
    - If the aggregated IMDNs received by the Messaging Server contain both in-network and inter-Service Provider Chat message senders, the Messaging Server will repackage the aggregated IMDNs according to in-network Chat message senders and inter-Service Provider Chat Message senders.

- Auto-acceptance of store and forward Messaging Server PUSH of stored notifications. Only the device which has sent the relevant message shall accept the notification;
- Store and forward Messaging Server PUSH of stored messages;
- An IMAP based Common Message Store expected to store OMA SIMPLE IM and CPM chat messages, described in [RCS5-CPM-MSGSTOR-ENDORS];
- Chat inactivity timeout: When a device or the network detects that there was no activity in a chat for IM SESSION TIMER, a configurable period of time (see Table 85), it will close the established Chat session;
- When reopening an older chat on the device, that contains messages for which a "display" notification should be sent, these notifications shall be sent as follows:
  - If there is no session established with the sender, the device will send the notifications outside a session (since there is no current session to send them to) using SIP MESSAGE;
  - If there is an active session but that session is with a device of the sender other than the one that was used to send the message to which this notification relates, the Messaging Server will ensure that these notifications are delivered outside of that session;
- The "IsComposing" notification is generated and processed according to the rules and procedure of [RCS5-SIMPLEIM-ENDORS] and [RCS5-CPM-CONVFUNC-ENDORS]. Consequently, the 'IsComposing' notification is not sent with CPIM headers, and as such a delivery and/or displayed notification cannot be requested.
- The transfer of files while a Chat session is taking place shall be performed in a separate session. Note that this is only at protocol level. From the user experience perspective, they should be able to transfer files whilst in chat. Messages over a maximum size (MAX SIZE 1-to-1 IM in section A.1.3.3) should be transferred using File Transfer or a Large Message mode standalone message. When a network does not allow multimedia within a chat (see MULTIMEDIA IN CHAT in section A.1.3.3), all multimedia messages shall be transferred using File Transfer or a Large Message mode standalone message.

3.3.4.1.1 Client Side Spam/Black List Handling

When receiving a message from a sender included in the Black List (i.e. a spam sender) the receiving client's/device's implementation shall:

- Terminate the transaction with a 486 BUSY HERE sent back to the sender.
- The receiver will still issue a delivery notification with status "delivered" which will be sent back to the sender.
- From the UI point of view, the receiver should not be notified on the reception of a message from a blacklisted sender and the message should be copied to the spam filter.

3.3.4.1.2 Personal Network Blacklists handling

NOTE: In the present section, the BPEF as described in section 2.15.1 may be provided by the Messaging Server itself.

When supported and enabled, the PNB described in section 2.15 are applied by the BPEF at both origination and termination of 1-to-1 chat invitation requests.

The following *resource-lists* from Shared XDMS are checked by the BPEF by comparing the URI values used in the request and in the list:

- on origination:

a) upon initiation of the 1-to-1 chat, the BPEF of the originator checks the '*rcs_pnb_outchat_blockedusers*' list to verify that the recipient is not among the blocked users for this request by comparing URIs contained in the list with the URI value of the Request URI of the SIP request.
b) If found, the BPEF shall reject the chat with a *403 Forbidden* with a warning header set to "122 Function not allowed" towards the user without forwarding the SIP INVITE to the recipient's network.

- on termination:
  a) the BPEF checks the *'rcs_pnb_chat_blockedusers*' list, to verify if the originator of the chat is among the blacklisted users by comparing the URIs contained in the list with the URI values of the P-Asserted-Identity header field of the SIP request.
  b) If the sender is among the blacklisted users, the BPEF returns a *403 Forbidden* with a warning header set to "122 Function not allowed" to the originator's network, without forwarding the SIP INVITE to the recipient.
  c) If the Common Message Store feature is supported, it stores the Session History folder data as defined in [RCS5-CPM-MSGSTOR-ENDORS] for the blocked chat invite event.

3.3.4.1.3 Chat abnormal interruption

If a device in a chat suffers an abnormal termination of the Chat session, for example loss of coverage, the "Send" button may be disabled. If the device determines that a message could not be sent (e.g. failed response or received no response), it shall inform the user that the chat message was not sent. If the TCP connection is lost, the client should re-send it in a new chat session once re-registered.

Note that if the Messaging Servers involved in the chat have implemented store and forward functionality, then the Messaging Servers shall be responsible for storing any messages received while a chat has been abnormally interrupted.

In temporary interruption cases, for example a device was out of network coverage but is now again within network coverage, the chat can be continued from the same conversation window. In this case a new session has to be established with a SIP INVITE request.

3.3.4.1.4 Store and Forward Mode

The store and forward functionality in the network is optional and it is up to each Service Provider to decide whether to deploy it.

The store and forward functionality requires a Messaging Server. There are three possible scenarios to fulfil the requirement for store and forward functionality:

1. Sender and receiver are on networks with a Messaging Server supporting store and forward: In this case the receiver's side Messaging Server has the responsibility to store and forward IMs which are not delivered. The sender's side Messaging Server has the responsibility of storing the delivered/displayed notifications if the sender is offline.
2. Only the sender is on a network with a Messaging Server supporting store and forward: The sender's side Messaging Server has the responsibility to store and forward Chat messages and/or delivered/displayed notifications if immediate delivery was not possible. As it is in the sender's network, the Messaging Server will not have information on when the receiver is online; therefore a retry mechanism is used. Note that it is the Service Provider's decision whether they provide store and forward for chat messages on behalf of the receiver who is in a different network that does not support store and forward.
3. Only the receiver is on a network with a Messaging Server supporting store and forward: The receiver's side Messaging Server has the responsibility to store and forward Chat messages and/or delivered/displayed notifications if they cannot be delivered. As it is at

the receiver's side, that Messaging Server will not have information on when the sender is online. Therefore a retry mechanism is used to store and forward notifications that could not be delivered right away. Note whether a Service Provider provides store and forward for delivered/displayed notifications on behalf of the sender who is in a different network that does not support store and forward is optional.

With the introduction of application messages making use of the chat session (see section 3.5.4.8 and 3.10.4), also the store and forward functionality for chat will have to deal with those content types. That shall be done as follows:

- When accepting a Chat session on behalf of a user, a Messaging Server shall indicate support for the application message content types that it supports (e.g. in the *a=accept-wrapped-types* SDP attribute that it provides in the SIP 200 OK Response.
- Storage of such content shall be as any other message content
- When a client comes online, forwarding shall be as follows:
  - When included as a body of a SIP INVITE request, for non-text content a dedicated Accept-Contact header field shall be added to the INVITE request carrying the IARI defined for the service corresponding to the included content type in section 2.6.1.1.2 with the *require* and *explicit* parameters.
  - If delivery fails with a SIP 480 response, the Messaging Server shall store the original message and forward it later (again including an *Accept-Contact* header field as for the initial forward) when another device of the user comes online.
  - The message shall also remain stored for later delivery to another device if when the application message needs to be forwarded in a session that is already established and the *a=accept-wrapped-types* attribute provided by the client that accepted the session didn't include support for the corresponding MIME type.

The Messaging Server stores undelivered messages for a period that is determined by local server policy. If at the end of this period the messages have not been delivered, the Messaging Server discards them. This applies to notifications as well as messages.

A dedicated configuration setting (IM CAP ALWAYS ON, see Table 85 in Annex A for further reference) is used to configure the client to allow sending messages to offline users.

NOTE: The procedure for Messaging Server to store the chat message when the participant is temporarily unavailable is described in [RCS5-SIMPLEIM-ENDORS] or [RCS5-CPM-CONVFUNC-ENDORS] based on the CHAT MESSAGING TECHNOLOGY configuration parameter defined in Table 85.

3.3.4.1.5 Delivering stored disposition notifications

To be able to deliver delivered/displayed notifications that were stored to a sender's device that has come online again, without disrupting the user experience, the Messaging Server supporting the store and forward functionality shall initiate a special session for the purpose of delivering these notifications. This special session shall be automatically accepted by the device. It is recognized by the device by means of the well-known username part of the URI (*rcse-standfw@<domain>*) uniquely identifying the store and forward service identity that is provided in the *P-Asserted-Identity* header field. Optionally an operator can disable the delivering of the stored notifications when the RCS user is roaming in a foreign network.

NOTE: The Messaging Server may also use Pager Mode messaging to deliver stored delivery and displayed notifications.

The Messaging Server supporting the store and forward functionality is required to send the delivered/displayed notifications to the exact device that has previously sent the associated messages. Therefore the Messaging Server implementing multidevice handling shall

support device identification as specified in section 2.11.3 (i.e. both GRUU and sip.instance support).

NOTE: The procedure for Messaging Server to deliver the stored chat messages and associated disposition notifications are described in [RCS5-SIMPLEIM-ENDORS] or [RCS5-CPM-CONVFUNC-ENDORS] based on the CHAT MESSAGING TECHNOLOGY configuration parameter defined in Table 85.

3.3.4.1.6 Interworking towards SMS/MMS

The functionality for interworking of the chat service to SMS/MMS is optional and it is the decision of each Service Provider whether to deploy it. This deployment involves

- the Messaging Server described in [RCS5-SIMPLEIM-ENDORS] or [RCS5-CPM-CONVFUNC-ENDORS]
- The ISF described in [RCS5-CPM-IW-ENDORS] which is responsible for selecting the appropriate interworking function for a new session
- The IWF for SMS and MMS described in respectively [RCS5-3GPP-SMSIW-ENDORS] and [RCS5-CPM-IW-ENDORS] which are responsible for doing the actual interworking (that is the protocol conversions) between RCS based chat and SMS or MMS

Based on service-level agreements (SLAs), interworking of chat may occur on the originating side or the terminating side, the same circumstances as for interworking of messages with SMS/MMS described in section 3.2.4.6. In brief, the interworking is initiated by the Messaging Server either based on local information it may have about the recipient, or when it receives one of the following error responses on the INVITE request that indicate that the recipient is not an RCS contact:

- 404 Not Found;
- 405 Method Not Allowed;
- 410 Gone;
- 414 Request URI Too Long;
- 415 Unsupported Media Type;
- 416 Unsupported URI Scheme;
- 488 Not Acceptable Here;
- 606 Not Acceptable.

3.3.4.1.6.1 Interworking at Originating Side

When a Chat session invitation needs to be interworked on the originating side, the CPM Participating Function will route the invitation to the ISF, which will select either SMS or MMS interworking based on applicable service provider policy. The ISF will then route the message to the selected IWF, which will either accept the chat invitation automatically on behalf of the SMS/MMS user, or will convert the chat invitation to an SMS/MMS invitation message and deliver it to the terminating network using the appropriate SMS/MMS NNI. The response to the chat invitation from the SMS/MMS user must be received through the same SMS/MMS interface to associate correctly the response with the earlier invitation. The SMS/MMS response (either accept or decline) to the invitation is converted to the appropriate SIP response and conveyed back to the RCS user.

3.3.4.1.6.2 Interworking at Terminating Side

When a Chat session invitation needs to be interworked on the terminating side, the invitation will be first routed to the terminating network as described in previous sub-sections, and then the same procedures as for interworking of chat invitations on the originating side will apply.

3.3.4.1.7 Multidevice handling

Multidevice handling occurs when a user has more than one device (e.g., PC and mobile).

When a new 1-to-1 chat is initiated and a message is sent from User A to a User B with User B having multiple devices registered at the same time, the network or Messaging Server forks the Chat session invitation to the different devices. Forking on the Messaging Server is further elaborated in section 3.3.4.1.7.1. Note that it is assumed that the originating user uses one device per session.

Each of User B's devices that receive the session invitation with a message in the INVITE generates a SIP MESSAGE request to carry the delivered IMDN. In a multidevice scenario, if a sender receives more than one IMDN for a sent message, it shall discard all copies except the first one it receives.

User B is able to respond to the chat from any of their devices. When they answer and send a message from one of the devices, that device (B1) becomes the only active device for User B and all the Chat sessions towards the other devices are terminated.

Once the user has answered the chat from device B1, all the subsequent messages sent to User B are received only by the active device B1 using the already established Chat session.

Device switching:

1. If User B closes the Chat session from the active device (either by closing the chat conversation from the chat window or due to an abnormal termination), any new messages sent by User A through the chat will make the Messaging Server establish the chat again using one Chat session per connected device of User B and send the message to them all.
2. If User B changes from one device B1 to another B2 by sending a new message to the chat from the new device B2, B2 will send a new INVITE request that will go to User A's device. When User A's device detects a new INVITE request from User B which already has an established session with User A's device it shall end that session and accept the new one. All subsequent messages are received only by device B2. Device B2 must then store the received messages and display them appropriately. If User A still has delivery and display reports for device B1, they should be sent before User A's device closes the old session.

When 1-to-1 chat messages are stored, the device identifier of the sender is stored with the message. When these messages are delivered, the device identifier for the sender of each message is not provided to the recipient. The Messaging Server that has stored the device identifier for the message intercepts each notification generated for that message and adds the device identifier so that it will reach the right device of the sender.

However, if there is a long delay before the notifications arrive (e.g., display notifications may arrive much later or may never arrive at all since a recipient is allowed to not send them), there is a risk that the correct device identifier may no longer be available in the Messaging Server. If this occurs, the notification will be sent to the sender with no device identifier at all, meaning if it is sent via SIP MESSAGE request only devices which are online will receive it, or if it is sent via a special session to deliver notifications, it will arrive at the device that first answers the SIP INVITE request, which might not be the right device.

Service Provider policy decides how long the Messaging Server shall keep device identifier information about stored messages after those messages have been delivered.

If the Messaging Server is an endpoint for a session with the sender of the stored messages when at the same time session is successfully set up to the recipient to deliver stored messages, the Messaging Server shall ensure that only notifications for the particular

device the sender is using are sent via MSRP towards the sender, while notifications for messages sent from other devices of the sender are each addressed to the correct device.

3.3.4.1.7.1 Forking on the Messaging Server

Forking to registered online devices in case that there is no message in the incoming INVITE request shall be achieved by using the forking capability at the Messaging Server. This capability shall be implemented on the terminating Participating Function.

As described in section 3.3.4.1.7, in case of an incoming INVITE request, User B is able to respond to the chat from any of their devices. When User B responds from one of the devices, that device becomes the only “active” device (i.e. ‘is composing’ notifications or messages originated from that device) for that user and, consequently, the terminating Participating Function shall tear down all other chat sessions towards the other devices under the same chat session identity by sending a SIP BYE request including a Reason header field with the protocol set to SIP and the protocol-cause set to 200 along with an optional protocol-text (e.g. SIP;cause=200;text="Call completed elsewhere"). A client may use this information to update its representation in the UI.

When a device belonging to User B registers in IMS and provided there is no other active device, the terminating Participating Function shall send an INVITE request for that chat session to the newly registered device.

3.3.4.1.8 Emoticons

Selected emoticons are displayed graphically but sent and received as text. The list of supported emoticons is defined in [RCS5-SIMPLEIM-ENDORS] Appendix N.

3.3.4.1.9 Chat message size limitations

The maximum size is controlled through the MAX SIZE 1-to-1 IM configuration parameter defined in Table 85. Messages that are larger than the maximum size indicated in the MAX SIZE 1-to-1 IM configuration parameter can be delivered either using File Transfer or a Large Message Mode standalone message.

3.3.4.1.10 Message Revocation

Message revocation is a feature that allows a client or Messaging Server to request for a chat message to be revoked by the recipient’s Messaging Server. The recipient’s Messaging Server processes MessageRevoke requests and responds with a Message Revoke Response request based on the chat message delivery status.

3.3.4.1.10.1 Generating Chat Message Revoke Requests

The MessageRevoke request is generated by either the client or the Messaging Server of the message sender, depending on the Service Provider policy. The MessageRevoke request is carried in the body of a SIP MESSAGE request that includes the same imdn.message-ID value of the chat message that is intended to be revoked (as described in section 3.3.4.1.10.4).

MessageRevoke requests shall be generated only towards networks where their Messaging Server can handle them as described in section 3.3.4.1.10.2 and are not meant to reach other clients. MessageRevoke requests shall not be generated in case the delivery notification pertaining to the original message has been received.

NOTE: Given that a revoke may be sent only if support has been indicated by the terminating network, it cannot be initiated when the INVITE transaction is still pending.

3.3.4.1.10.1.1 Message Revoke Requests by the Client

When message revocation is enabled (CHAT REVOKE TIMER, see section A.1.3.3), the client can generate MessageRevoke requests once the timer is expired. In order for the MessageRevoke requests to be transmitted, the client shall have data connectivity.

MessageRevoke requests shall be generated only if support for MessageRevoke requests has been indicated in the Contact header of the SIP INVITE request or in the response to the SIP INVITE request of the session to which the message intended to be revoked belongs. Specifically, this indication is in the form of a feature tag in the Contact header of the SIP INVITE request or response that is defined in section 3.3.4.1.10.3. For session initiation scenarios that result in SIP *486 Busy Here* responses from the terminating Messaging Server, a feature tag cannot be included by the terminating network since according to [RFC3261] a Contact header is not present in such responses. However, depending on Service Provider policy a MessageRevoke request may be generated (for the case where a first message is included in the SIP INVITE request). In that case, this MessageRevoke request might be blocked on the NNI if not supported by the terminating network (see section 3.3.5.4).

When a message is to be revoked, the client shall include an Accept-Contact header field with either the CPM ICSI for Session Mode Messaging, or the OMA SIMPLE IM feature tag (depending on the value of the CHAT MESSAGING TECHNOLOGY configuration parameter defined in Table 85), as is already the case for IMDNs carried in SIP MESSAGE requests, and shall add a dedicated Accept-Contact header field carrying the Message Revoke feature tag defined in section 3.3.4.1.10.3 along with the *require* and *explicit* parameters. The client shall also include the message revocation content-type including the value of the imdn.message-ID of the original message that is requested to be revoked, as described in section 3.3.4.1.10.4. The Request-URI of the MessageRevoke request shall be set to the address of the target contact of the message that is requested to be revoked. The body of the MessageRevoke request, as described in section 3.3.4.1.10.4, shall have:

- the <Message-ID> element set to the value of the imdn.message-ID of the original message that is requested to be revoked,
- the <From> element set to the URI of the sender of the message
- the <To> element set to the URI of the recipient of the message

3.3.4.1.10.1.2 Message Revoke Requests by the Messaging Server

Similarly to section 3.3.4.1.10.1.1, MessageRevoke requests shall be generated only if support for MessageRevoke requests has been indicated in the Contact header of the SIP INVITE request or in the response to the SIP INVITE request of the session to which the message intended to be revoked belongs. Specifically, this indication is in the form of a feature tag in the Contact header of the SIP INVITE request or response. For session initiation scenarios that result in SIP *486 Busy Here* responses from the terminating Messaging Server, a feature tag cannot be included by the terminating network. However, depending on Service Provider Policy, a Message Revoke request may be generated (for the case where a first message is included in the SIP INVITE request). In that case, this MessageRevoke request might be blocked on the NNI if not supported by the terminating network (see section 3.3.5.4).

The format of the MessageRevoke request generated by the Messaging Server is the same as described for the MessageRevoke requests by the Client in section 3.3.4.1.10.1.1.

3.3.4.1.10.2 Handling MessageRevoke Requests

For a network, handling of MessageRevoke requests goes along with having the functionality to generate MessageRevoke requests (either by the client or by the Messaging Server) and vice versa.

The MessageRevokeResponse request shall indicate the result of the MessageRevoke request that can be either successful or failed. Similarly to the MessageRevoke request, the MessageRevokeResponse request include an Accept-Contact header field with either the CPM ICSI for Session Mode Messaging, or the OMA SIMPLE IM feature tag (depending on the value of the CHAT MESSAGING TECHNOLOGY configuration parameter defined in Table 85), as it is already the case for IMDNs carried in SIP MESSAGE requests, and shall add a dedicated Accept-Contact header field carrying the Message Revoke feature tag defined in section 3.3.4.1.10.3 without the *require* and *explicit* parameters. The Messaging Server handling the MessageRevoke request shall also include the message revocation content-type including the value of the imdn.message-ID of the original message that was requested to be revoked, and the revoke result parameter as described in section 3.3.4.1.10.4. The MessageRevokeResponse request shall include the device identifier (GRUU or sip.instance feature tag) as specified in section 2.11.3 based on the device identifier included upon chat session establishment. The Request-URI of the MessageRevokeResponse request shall be set to the address of the contact that sent the message that is requested to be revoked.

The MessageRevokeResponse request shall be indicated as successful when the message to be revoked is removed from the deferred storage and will therefore not be delivered to the client.

The MessageRevokeResponse request shall be indicated as failed when any of the following conditions is met:

- Interworking towards SMS/MMS has occurred at originating or terminating side
- A successful delivery notification for which the MessageRevoke request has been generated has been received by the originating or terminating Messaging Server;
- Message revocation is not performed successfully by the terminating Messaging Server (e.g., due to Messaging Server failures);
- The message that the intended MessageRevoke request has been generated for is stored at the terminating side in the Common Message Store.

MessageRevoke requests shall never be forwarded to the client and shall be processed right after being received by the Messaging Server.

3.3.4.1.10.3 Message Revoke feature tag

RCS defines a Message Revoke feature tag to indicate support of the message revocation feature. The RCS Client and originating Messaging Server shall make use of the message revocation feature only when the terminating Messaging Server has indicated its support through the Message Revoke feature tag. It can be used to indicate support for revoking of any message identified with a CPIM Message-ID. However, this release of RCS only allows it for chat messages.

The feature tag is set in the Contact header of the SIP INVITE request or response used to set up a 1-to-1 chat session and it is always attached by the Messaging Server that supports Message revocation feature. The client shall only include this feature tag in the MessageRevoke request.

The feature tag is defined as *+g.gsma.rcs.msgrevoke*.

3.3.4.1.10.4 Message Revoke content-type

The Message Revoke XML schema is defined as shown on Table 54.

The associated MIME content type is *application/vnd.gsma.rcsrevoke+xml*.

This content type used in both the MessageRevoke request and in the MessageRevokeResponse request.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:gsma:params:xml:ns:rcs:rcs:rcsrevoke"
                xmlns:xs="http://www.w3.org/2001/XMLSchema"
                xmlns="urn:gsma:params:xml:ns:rcs:rcs:rcsrevoke"
                elementFormDefault="qualified"
                attributeFormDefault="unqualified">
        <xs:element name="imRevoke">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="Message-ID" >
                                        <xs:simpleType>
                                                <xs:restriction base="xs:token"/>
                                        </xs:simpleType>
                                </xs:element>
                                <xs:element name="result" minOccurs="0" maxOccurs="1">
                                        <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                        <xs:enumeration value="success"/>
                                                        <xs:enumeration value="failure"/>
                                                </xs:restriction>
                                        </xs:simpleType>
                                </xs:element>
                                <xs:element name="From">
                                        <xs:simpleType>
                                                <xs:restriction base="xs:anyURI"/>
                                        </xs:simpleType>
                                </xs:element>
                                <xs:element name="To">
                                        <xs:simpleType>
                                                <xs:restriction base="xs:anyURI"/>
                                        </xs:simpleType>
                                </xs:element>
                                <xs:any namespace="##other" processContents="lax" minOccurs="0"
                                maxOccurs="unbounded"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
</xs:schema>
```

Table 54: RCS Revoke and RevokeResponse message body schema

The following is an example of the body of a SIP MESSAGE requesting that a specific chat message be revoked. In order to know whether the revoke was successful or not, the MessageRevoke request sender checks the result field in incoming MessageRevokeResponse requests.

Example of a MessageRevoke request:

```
Content-type: application/vnd.gsma.rcsrevoke+xml
Content-length: ...

<?xml version="1.0" encoding="UTF-8"?>
<imRevoke xmlns="urn:gsma:params:xml:ns:rcs:rcs:rcsrevoke">
<Message-ID>23499fuq34fu</Message-ID>
<From>tel:+1234578901</From>
<To>tel:+1234578902</To>
</imRevoke>
```

Note that the Message-ID, "23499fuq34fu", in the XML body refers to the CPIM Message-ID of the message to be revoked.

Example of a MessageRevokeResponse request where the revoke succeeded. If it had failed, the value of result would be “failed”:

```
Content-type: application/vnd.gsma.rcsrevoke+xml
Content-length: ...

<?xml version="1.0" encoding="UTF-8"?>
<imRevoke xmlns="urn:gsma:params:xml:ns:rcs:rcs:rcsrevoke">
<Message-ID>23499fuq34fu</Message-ID>
<result>success</result>
<From>tel:+1234578901</From>
<To>tel:+1234578902</To>
</imRevoke>
```

Note that the Message-ID, "23499fuq34fu", in the XML body refers to the CPIM Message-ID of the message that was revoked.

3.3.4.2 Technical Realization of 1-to-1 Chat features when using OMA SIMPLE IM

At the technical level the 1-to-1 Chat service implemented using OMA SIMPLE IM extends the concepts described in section 3.3.4.1 with the following concepts:

- For OMA SIMPLE IM, first message is always included in a CPIM/IMDN wrapper carried in the SIP INVITE request. So the configuration parameter FIRST MSG IN INVITE defined in Table 85 is always set to 1. A client should always include “positive-delivery” in the value for the Disposition-Notification header field in that message. That means that the value of the header field is either “positive-delivery” or “positive-delivery,display” depending on whether display notifications were requested. The value of “negative-delivery” is not used in RCS for 1-to-1 Chat. SIP INVITE requests for a one-to-one session that carry a message in CPIM/IMDN wrapper shall be rejected by the server unless they carry a Disposition-Notification header that at least includes “positive-delivery”.
- If auto-accept is not used, then the devices each send a SIP 180 response toward A.
- The received Chat session invitation contains an IMDN requesting 'delivery' notification. So each receiving device sends back a SIP MESSAGE request containing the IMDN indicating successful delivery of the original message sent by A.
- The receiving clients each send a 486 BUSY HERE response to the outstanding INVITE when a new INVITE arrives from the same user so that there is not more than one outstanding INVITE from one user. The IMDN for 'delivery' notification is requested and sent similarly to the first session invitation.
- No support for exchanging multimedia content within a chat so the configuration parameter MULTIMEDIA IN CHAT defined in Table 85 is always set to 0 in order to reduce the complexity associated with the store and forward functionality. Therefore in the SDP of the SIP INVITE request and response, the *a=accept-wrapped-types* attribute

shall only include text/plain and message/imdn+xml. If File Transfer using HTTP is supported (see section 3.5.4.8) then the *a=accept-wrapped-types* attribute shall also include *application/vnd.gsma.rcs-ft-http+xml*. If Geolocation PUSH is supported (see section 3.10.4.1.3), then the *a=accept-wrapped-types* attribute shall also include *application/vnd.gsma.rcspushlocation+xml*. To transfer multimedia content during a chat, File Transfer is used.

- When one of User B's devices detects user activity relevant to the consumption of the message contained in the invitation (e.g. click on a pop-up to go to the Chat window) a 1-to-1 chat session is established according to the following possible criteria:
  a) The respective client returns a 200 OK response, signalling the initiation of the remaining procedures to establish the chat when User B reacts to the notification by opening the chat window. This is the default criteria for RCS 5.1 and, consequently, all the diagrams shown in this document reflect this behaviour.
  b) The 200 OK response is sent when User B starts to type a message, or
  c) The 200 OK response is sent when User B sends a message. Please note that in this case User B's message will not generate an invite but is buffered in the client until the MSRP session is successfully established.
  d) The 200 OK response is sent immediately since the devices receiving the invitation are configured to auto-accept the session invitations (IM SESSION AUTO ACCEPT configuration parameter defined in Table 85).

  Please note that:

  - The behaviour for criteria a), b) and c) is configured via the IM SESSION START parameter as defined in Table 85. The behaviour for criteria d) is configured via the IM SESSION AUTO ACCEPT configuration parameter defined in Table 85.
  - For a), b) and c), the 200 OK is sent if the chat invitation has not expired. Otherwise, User B's message shall be sent in a new invitation (from User B to User A).

  If the Chat session invitation from User A contained an IMDN *Disposition-Notification* header requesting a 'display' notification and if the privacy settings allow it, the device User B is using shall generate an MSRP SEND request toward User A that contains the IMDN 'display' status for the message received from User A.

  It may be the case that multiple Chat sessions from User A are pending on User B's side, that is the last received Chat session is established and the other pending sessions are answered with a 486 BUSY HERE response. In such cases, if the Chat session invitations from User A contained a IMDN Disposition-Notification header requesting a 'display' notification, the device of User B that accepted the SIP INVITE generates an MSRP SEND request toward User A that contains the IMDN 'display' status for each message received from User A.

  NOTE: The statement in section 3.3.4.1 that the CPIM/IMDN wrapper shall be *UTF-8* encoded to avoid any potential internationalization issues also applies to the IMDN requested in the SIP INVITE request.

- A Messaging Server supporting store and forward behaves as a back-to-back user agent handling the SIP INVITE requests that are used to establish the chat session. While doing this it may have to return a different response to the INVITE request on the originating leg than the one it received on the INVITE request on the terminating leg. The mappings shown in Table 55 will be applied:

Response received on terminating leg	Response sent on originating leg	Store the message
480 Temporarily unavailable	200 OK	Y
408 Request Timeout	486 Busy Here	Y
487 Request Terminated	486 Busy Here	Y
500 Server Internal Error	486 Busy Here	Y
503 Service Unavailable	486 Busy Here	Y
504 Server Timeout	486 Busy Here	Y
600 Busy Everywhere	486 Busy Here	Y
603 Decline	486 Busy Here	Y
Any other response (including 404 Not Found and 200 OK)	Received response (that is no mapping is done)	N

Table 55: Mapping of received Error Responses by the Messaging Server

- To reduce the complexity at protocol level and avoid potential TCP switchover(s), it is recommended to limit the maximum size of a chat message (see section 3.3.4.1.9) to avoid the SIP INVITE request to be longer than the path MTU (e.g., 1300 bytes) and, consequently, trigger the TCP switchover. The maximum size controlled through the MAX SIZE 1-to-1 IM configuration parameter defined in Table 85 applies to both the first message in the INVITE and to messages sent via MSRP. If the user attempts to send a first or subsequent chat message larger than this limit (counting the size of the CPIM body only, that is CPIM headers are not included in the count), then the user shall be notified that the message is too large.
- In the first message in the INVITE, the client shall set both the CPIM From and CPIM To headers to *sip:anonymous@anonymous.invalid* to prevent revealing the user's identity when transmitted over unprotected links. A client receiving a CPIM message in a one-to-one Chat should therefore ignore the identity indicated in the CPIM headers.

3.3.4.2.1 Clarifications on Chat race conditions

- Two simultaneous invites. Though unlikely, it may be possible that two users decide to invite each other simultaneously for a chat. In this situation the behaviour of the clients should be the following:
  - User A sends an invite to User B for Chat
  - Before a final response for that invite is received, User A receives an invite from User B for Chat
  - User A will send a 486 BUSY HERE response to User B. In addition to this, User A will send the correspondent delivery and read notification using SIP MESSAGE.
  - From the UX point of view, the message sent by B will be displayed as received.
  - User B will behave as user A, potentially resulting in both session invitations being turned down with a SIP 486 BUSY HERE response. Users will have to retry session setup until successful.
- New invite sent after a previous invite has been accepted. Though unlikely, the following scenario can take place:
  - User A sends an invite for chat to User B
  - User B accepts the chat a 200 OK response is sent back to User A

  - In parallel and before receiving the 200 OK response, User A sends a new invite with a new message
- To resolve the race condition:
  - When User B receives the new invitation, it should terminate the current MSRP session (if established) by sending a SIP BYE
  - Once the initial session is terminated, a new 200 OK response should be issued which will trigger the establishment of a new MSRP session.

For additional clarification, explanatory diagrams have been included in Annex B, sections B.1.9 and B.1.10.

3.3.4.3 Technical Realization of 1-to-1 Chat features when using OMA CPM

At a technical level the Chat service implemented using OMA CPM extends the concepts described in section 3.3.4.1 with the following concepts:

- It is an UNI implementation choice to include the first chat message in the CPIM/IMDN wrapper of the SIP INVITE request through the setting of the FIRST_MSG_IN_INVITE configuration parameter in Annex A. The RCS NNI shall follow the OMA CPM v1.0 standards, which does not carry user message in the SIP INVITE.
- If a Service Provider expects that the first chat message be sent after the session is established, the configuration parameter FIRST MSG IN INVITE in Annex A is set to disabled.
  Please note that this is the recommended approach when using OMA CPM as this secures compatibility to the existing standards.
- If a Service Provider expects that the first message be in a CPIM/IMDN wrapper of the SIP INVITE request, the configuration parameter FIRST MSG IN INVITE defined in Table 85 is set to 1. A client should always include “positive-delivery” in the value for the Disposition-Notification header field in that message. That means that the value of the header field is either “positive-delivery” or “positive-delivery,display” depending on whether display notifications were requested. The value of “negative-delivery” is not used in RCS for 1-to-1 Chat. SIP INVITE requests for a one-to-one session that carry a message in CPIM/IMDN wrapper shall be rejected by the server unless they carry a Disposition-Notification header that at least includes “positive-delivery”.
- If the first message is included in a CPIM/IMDN wrapper in the SIP INVITE request, the client shall set both the CPIM From and CPIM To headers of that wrapper to *sip:anonymous@anonymous.invalid* to prevent revealing the user’s identity when transmitted over unprotected links. A client receiving a CPIM message in a one-to-one Chat should therefore ignore the identity indicated in the CPIM headers.
- If auto-accept is not used, then the devices send a SIP 180 response toward A.
- When users are allowed to have multiple devices and those devices are configured to auto-accept (IM SESSION AUTO ACCEPT set to 1, as defined in section A.1.3.3), the Messaging Server is required to be able to fork the incoming 1-to-1 Chat session request to each of the receiving user's devices to set up an MSRP session with each of them.
- The receiving clients (or their Participating Function on their behalf) each send a 486 BUSY HERE response to the outstanding INVITE request when a new INVITE request arrives from the same user so there is not more than one outstanding INVITE request from one user.
- If multimedia content within a Chat session is permitted the configuration parameter MULTIMEDIA IN CHAT defined in Table 85 is set to 1. Therefore in the SDP of the SIP INVITE request and response, the *a=accept-wrapped-types* attribute shall only include either *, or a complete list of all content types supported during the Chat session

(including at least text/plain and message/imdn+xml), e.g., *a=accept-wrapped-types:**.

Note that if the SIP INVITE request is allowed to carry the first chat message in a CPIM/IMDN wrapper (that is, the FIRST MSG IN INVITE configuration parameter defined in Table 85 is set to 1) and the first chat message contains multimedia content, then the chat message shall not be included in the SIP INVITE request. It will be sent via MSRP once the session is established (e.g. the auto-accept implementation). If the chat session is not established, the multimedia content chat message may be sent via File Transfer or Large Message Mode (see section 3.2) depending on the receiver's capabilities. If both sender and receiver support both File Transfer and Large Message Mode, the client should use Large Message Mode. .

- If multimedia content within a Chat session is not a permitted, the configuration parameter MULTIMEDIA IN CHAT defined in Table 85 is set to 0. Therefore in the SDP of the SIP INVITE request and response, the *a=accept-wrapped-types* attribute shall only include text/plain and message/imdn+xml and if File Transfer using HTTP or Geolocation PUSH is supported (see sections 3.5.4.8 and 3.10.4.1.3) *application/vnd.gsma.rcs-ft-http+xml* and *application/vnd.gsma.rcspushlocation+xml* respectively, e.g., *a=accept-wrapped-types:text/plain message/imdn+xml*. To transfer multimedia content during a chat, File Transfer is used.
- When one of User B's devices detects user activity relevant to the consumption of Chat session invitation (e.g. click on a pop-up to go to the Chat window) a 1-to-1 chat session is established according to the following possible criteria:
  a) The respective client returns a 200 OK response, signalling the initiation of the remaining procedures to establish the chat when User B reacts to the notification by opening the chat window. This is the default criteria for RCS 5.1 and, consequently, all the diagrams shown in this document reflect this behaviour.
  b) The 200 OK response is sent when User B starts to type a message, or
  c) The 200 OK response is sent when User B sends a message. User B's message is buffered in the client until the MSRP session is successfully established.
  d) The 200 OK response is sent immediately if the devices receiving the invitation are configured to auto-accept[25] the session invitations (IM SESSION AUTO ACCEPT configuration parameter defined in Table 85).

Please note that the behaviour for criteria a), b) and c) is configured via the IM SESSION START parameter as defined in Table 85. The behaviour for criteria d) is configured via the IM SESSION AUTO ACCEPT configuration parameter defined in Table 85.

3.3.4.3.1 1-to-1 Chat Service Identification

The RCS client shall populate the P-Preferred-Service header field in all CPM requests with the CPM Feature tag defined for the service, as described in [RCS5-CPM-CONVFUNC-ENDORS]. The S-CSCF or AS that performs the service assertion in the originating network shall add the P-Asserted-Service header field set to the value of the asserted CPM service ICSI (i.e. "urn:urn-7:3gpp-service.ims.icsi.oma.cpm.session" for CPM chat, or "urn:urn-7:3gpp-service.ims.icsi.oma.cpm.deferred" for deferred delivery done as part of the Store and forward realization) and remove the P-Preferred-Service header field before further routing the request.

A receiving network element and RCS client should ignore any SIP header fields that they do not understand (e.g. P-Preferred-Service, or P-Asserted-Service header fields).

[25] Note that the Service Provider multidevice policy has to be consistent with Chat auto-acceptance policy.

3.3.4.4 Common Message Store

The procedures defined in section 3.2.4.7 also apply for 1-to-1 chat messages that may have been received via legacy means. The same correlation procedures apply.

3.3.5 NNI and IOT considerations

3.3.5.1 Chat session interworking when one side carries a message in the INVITE request

Interworking from a Chat session with a chat message in the INVITE request to a Chat session where the INVITE request does not carry any chat message requires that the Messaging Server (or a separate network entity) performing the interworking store the message in the INVITE until the Chat session without first message in INVITE is set up. If multiple Chat session INVITEs with chat messages arrive before the Chat session on the other side is set up, multiple chat messages are stored, however it is recommended that the Messaging Server automatically accept the session on behalf of a user in a network not supporting first message in the INVITE request. If no Chat session is set up on the other side, the chat messages are kept and delivery is attempted at a later time in the same way as already specified in section 3.3.4.1.4 when chat messages are stored on the originating side.

Interworking from a Chat session without first message in INVITE to a Chat session with a message in the INVITE requires that the Messaging Server accept the Chat session without any message on behalf of the recipient user and once the first chat message is received via MSRP, initiate an INVITE towards the recipient, including the first chat message as a CPIM body in the INVITE. Providing the recipient, or the recipient's Messaging Server on behalf of the recipient, does not set up a session, the Messaging Server performing the interworking continues to generate INVITEs towards the recipient for each new chat message received.

See the flows in Annex B for more information.

3.3.5.2 Interworking between a Chat session not allowing multimedia content and a Chat session allowing multimedia content

To allow interworking between a Chat session not allowing multimedia content, and a Chat session allowing multimedia content, SDP negotiation of the types of media and wrapped media allowed in the Chat session shall be used as specified in [RFC4975], and shall be respected by the devices and other endpoints involved in the session.

This occurs in NNI situations when one user is served by a network supporting multimedia content in Chat sessions and another user is served by a network supporting text only in Chat sessions. A Messaging Server where only text is allowed in Chat sessions shall ensure that the *a=accept-wrapped-types* attribute in the SDP used to negotiate use of MSRP only contains the *text/plain* and *message/imdn+xml* content-types and if File Transfer using HTTP or Geolocation PUSH is allowed (see sections 3.5.4.8 and 3.10.4.1.3) *application/vnd.gsma.rcs-ft-http+xml* and *application/vnd.gsma.rcspushlocation+xml* respectively.

See the flows in Annex B for more information.

3.3.5.3 SIMPLE IM session and CPM session interworking of feature tags

The mapping of the appropriate SIMPLE IM session feature tags is done as per Appendix G in [RCS5-CPM-CONVFUNC-ENDORS] when it is determined that the remote network requires such interworking. Also once a session is set up with the recipient, the Messaging Server or a separate network entity performing the interworking ensures that messages exchanged via MSRP are sent end to end.

See the flows in Annex B for more information.

3.3.5.4 Interworking between a service provider that supports "Message Revocation" and a service provider that does not support

To prevent the MessageRevoke request being leaked to the service provider's side that does not support message revocation, leading to unwanted user experience and possible undesired charging implication, the service provider that implements the message revocation shall make sure that the MessageRevoke requests are only sent to the service providers that also implement the message revocation when interworking.

3.3.6 Implementation guidelines and examples

Please note that where the specification describes the user interface, it should be taken as guidance.

3.3.6.1 General

End to end flows for store and forward 1-to-1 chat with notifications can be found in Annex B.

The following sections show the relevant chat message flows and reference user experience. Please note that the following assumptions have been made:

- For simplicity, the internal mobile network interactions are omitted in the diagrams shown in the following sections.
- Each Service Provider may deploy a Messaging Server (that is the use of a Messaging Server is optional in RCS deployments), to manage all messages from its customers.
- Prior to the chat, the user will have accessed their address book or Chat application to start the communication. As described previously, while these actions are performed an OPTIONS or Presence request is sent to verify the available capabilities. In the following diagrams it is assumed that this exchange (OPTIONS/Presence request and response) has already taken place, and therefore, both ends are aware of the capabilities and the available RCS services of the other side. If that is not the case, the OPTIONS (or Presence) request should be sent at the same time the chat is being set up.

Service Provider support of the store and forward functionality is optional in RCS. To allow a Service Provider to provide store and forward functionality to its customers even in cases where the Chat session is established towards a user of a Service Provider that does not support store and forward, the messages can optionally be stored and forwarded from the sender's Messaging Server, based on operator's policy.

If the Common Message Store is available, the device should synchronize with the Message Store Server when a user is about to initiate a chat with another user. Note, however, that this may not be desirable in roaming scenarios. As for Standalone Messaging, the same clarifications on the procedures at the Message Store Client (including synchronization) as described in section 3.2.6.2 apply for 1-to-1 Chat.

3.3.6.2 Entry points to the chat service

From the UX perspective there are three possible entry points to this service:

1. Address book/Call-log: chat can be initiated to any RCS contact with Chat capability as described in section 2.6.

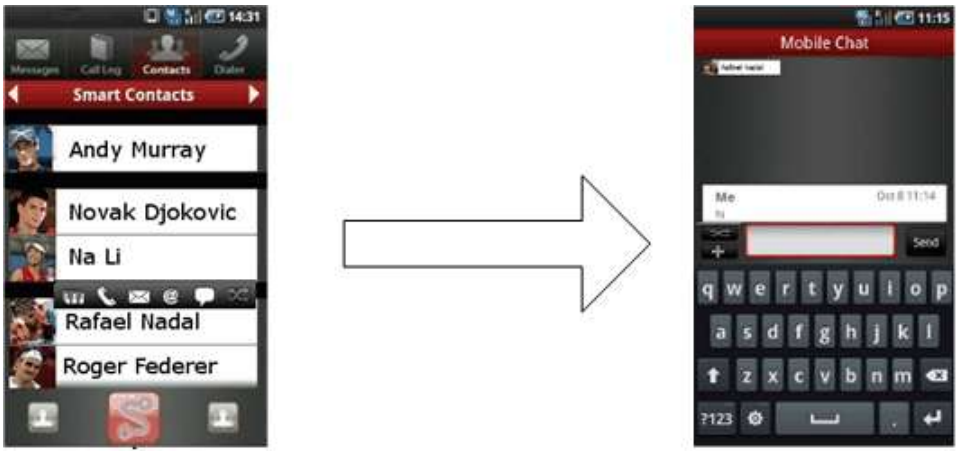


Figure 45: Reference UX for accessing chat from address book/call-log

2. Chat application: There should be a dedicated Chat application entry point in the device menu – task oriented initiation. This application will provide access to the chat history and gives the possibility to start a new chat.

Figure 46: Reference UX for starting a chat from the Chat application

Once the Chat application is opened, the user is presented with the complete list of RCS contacts with Chat capability. Whether or not contacts which are currently not registered are shown depends on the Chat store and forward policy (see IM CAP ALWAYS ON in Table 85) chosen by the Service Provider.
In addition to the "start a new chat" functionality, the Chat application allows the user to browse the Chat History, both 1-to-1 and Group Chat sessions:

Figure 47: Reference UX for starting chat from the Chat application history

In this case, when the chat is started the last messages exchanged with that contact (or group of contacts) are shown even though the conversation might have been from

another device. In the chat history the user can also browse through chat sessions that he has selected for permanent storage (if the Common Message Store feature is available for the user) and start a conversation from those. The context of the past chat is not relayed to the other party/parties.

3. File Transfer (receiver): When transferring a file and with the aim of establishing a communication context for the transfer (the receiver may want to know for instance why the sender is sharing that file), after the transfer has been accepted the file transfer is presented to the receiver as a chat UX with a file being transferred. Please note that at the time the File Transfer request is presented, the chat session is not started; the chat session will only start when/if the receiver sends a chat message back to the sender.

Figure 48: Reference UX for File Transfer on the receiver side

From the UX point of view the requirements are summarized below:

- The user shall not be aware of the solution which is being used to transfer the file both at originator and terminating side.
  - Sender: There is no difference from a UX point of view between an accepted transfer and the upload to the HTTP content server. A progress bar is shown to track the progress. Whether the user is warned when initiating a File Transfer to an offline recipient depends on the same IM WARN SF setting that is used for Chat.
  - Recipient: There is also no difference from a UX point of view in terms that the accept button is shown and if pressed the file is either downloaded from the sender (accepting the SIP INVITE) or from the content server (via HTTP/HTTPS)
- The only addition to the experience is the delivery notification that shall show the user when the file has been received by the other party (not stored in a middle-man server). In terms of UI, it is recommended to reuse the approach that is already in place for chat notifications so the UX is consistent among services.

Please refer to section 3.5 which covers the RCS File Transfer service in detail.

3.3.6.3 Initiating a chat

RCS User A initiates a chat by selecting one of his contacts User B from the address book, contact list or Chat application in one of his devices.

The device of User A determines whether User B is available to use the Chat service at that time, using one of the methods in section 2.6.

If User B is not available and there is no Chat store and forward Server on User A's side nor on User B's side, or no chat interworking to SMS/MMS, or if an answer to the query is not received in less than a time lapse (left to OEM User Experience criteria), then the contact is shown as ‘Not available for a Chat session’, and the SMS/MMS service or CPM Standalone Messaging service could be offered as a messaging option. Once the availability of the chat

service is ensured end-to-end and User A performs the appropriate UI actions on the device, a message composer and an empty chat window are opened.

When User A types the first message and presses the “Send” button, device A will initiate a Chat session invitation toward B (for the multidevice scenario see multidevice handling in section 3.3.4.1.7).

The one or more devices of User B receiving a Chat session invitation, may either all be configured to auto-accept the invitation, or the devices may wait for user action before accepting the invitation. If a spam filter or a black list is implemented on any one of User B’s receiving devices and User A is in the black list, the invitation is terminated following the procedure described in section 3.3.4.1.1.

On the User B side, a notification (UI dependent) is displayed on each receiving device to inform the user about the incoming message. The user is able to read the message and go to the chat window to answer the message on the device of his choice.

User A can type additional messages before the chat is answered, that is before the Chat session is established. On User B's side, a notification may be displayed for each received message (UI dependent). The sender’s device may buffer the messages if the device is configured to wait for a Chat session to be set up before sending messages.

3.3.6.4 Answering a chat

There may or may not be an explicit acceptance of the user to answer a chat.

3.3.6.5 Messages exchanged in an established chat

Providing a Chat session is established, messages are exchanged between User A and User B. A delivery notification is requested for each message and a display notification is optionally requested.

The recommendation is to show the information received in the delivery and display notifications only within the Chat window without the need for a pop-up or information message when the user is outside of the Chat application.

3.3.6.6 Message display and message store

All messages are stored in the participating devices, together with a time indication and an appropriate indication of the sender and the receiver of each message. This time indication shall be obtained from the CPIM *DateTime* header for received messages. Since according to section 3.3.4.1 these values should be set by the Messaging Server, this allows for a correct time based indication for those messages without depending on the device’s own clock which may not have been set correctly. For sent messages however the only clock available at transmission time is the device’s own clock.

However, it is Messaging Server responsibility to deliver messages in the correct order, so the RCS Client is able to rely on the reception time in order to interleave the incoming and outgoing messages. Please note that the ordering of the messages is phone clock based, the shown message time at the UX should be the network time (when available) in order to correctly display the time of store and forwarded messages.

When a Common Message Store is available for the user, the messages are synchronized with the Message Store Server as specified in [RCS5-CPM-MSGSTOR-ENDORS].

When the storage limit is reached, deletion might occur on a first in/first out (FIFO) queue policy. It is open to OEM criteria how to implement other opt-in deletion mechanisms (e.g., ask always, delete always, delete any conversation/message from specific contacts, etc.).

3.3.6.7 Leaving the chat composing window

Once a 1-to-1 chat is established any of the two users can leave the composing window without closing the chat. For example, a user could move to his mobile home screen to check an incoming email, or make a phone call.

While the chat composing window is not shown (that is, it is not the foreground window) any incoming message belonging to that chat will trigger a status notification (UI dependent) so the user is aware of the new message and, may return to the chat composing window to answer it.

Also, the user could decide to return to the chat composing window and send a new message without receiving one. The user would be able to achieve this via the Chat application, which will display the ongoing chats, or via the address book by clicking on the contact with whom he is involved in the chat session.

In both cases, when the user returns to the chat composing window, all the messages are displayed.

3.3.6.8 'IsComposing' notification

When a user starts typing in the chat composition window and privacy settings allow it, an 'IsComposing' notification is sent to the other user. That user's UI will then display an indication in the chat composing window to indicate it (UI dependent).

The recommendation is to show the information received in the 'IsComposing' notification only within the Chat window without the need for a pop-up or information message when the user is outside of the Chat application.

The 'IsComposing' indication is removed from the UI when a new message is received, when a timeout occurs without receiving a new message, or when a new 'IsComposing' notification arrives.

3.3.6.9 Closing a chat / Re-opening a chat

Any of the two users can close the Chat session. This can be achieved from the chat composing window or from the Chat application.

The user should be able to re-open the chat. However the resulting action at protocol level would depend on whether the Chat session is still open or not.

Closing the Chat session may not be notified to the remote user in the chat. At protocol level, the session is terminated. Therefore if the remote user sends a message, a process similar to the initiation of a Chat session is performed as described in 3.3.6.3.

3.3.6.10 Re-Opening an older chat

An old chat conversation can be re-opened. From the user perspective, it is the same procedure as for initiating a chat (see section 3.3.6.3), except that when a message is sent, a new Chat session is established.

The device will then display the previously stored conversations with that contact preceding the current active one. If any displayed notifications still need to be generated, they are sent towards the original message sender.

3.3.6.11 User experience regarding notifications when several store and forward messages arrive in a short period of time

If a user has several chat messages waiting in storage in the Messaging Server to be delivered, the UX may be impacted if many Chat message notifications appear at the sender's device when the messages are delivered to the receiver after the receiving user gets registered again.

To avoid this situation and, specifically, when receiving stored and forwarded messages, the suggested experience follows:

- Only the delivery and/or display of the first message is shown in a notification to the sending user. The remaining store and forward messages are delivered but they do not cause a notification to be shown to the message sender.
- If messages from several sending users are received, only one message notification per user containing the first delivered message per sender is shown to the recipient user.

As mentioned previously this is a suggested guide and not mandated behaviour.

NOTE: the described behaviour refers to notifications shown on screen to the message recipient and does not affect the behaviour with regard to the sending of delivery notifications. Those are still sent for all received messages for which such a notification was requested.

3.4 Group Chat

3.4.1 Feature description

The Group Chat service enables users to exchange messages between many users instantly.

The following RCS features are described:

- Interworking of participants in a Group Chat to SMS/MMS
  This feature requires a Messaging Server to interwork the messages for participants without an RCS device to and from SMS or MMS.
- "Delivered" message disposition
  This allows the sender of a message to be notified when a message has been delivered to the recipient. These can be delivered inside and outside of the group chat.
- "Displayed" message disposition
  This allows the sender of a message to be notified when a message has been displayed on one of the recipient’s devices. Note that this notification cannot certify that the recipient has actually read the message. It can only indicate that the message has been displayed on the recipient's terminal User Interface (UI). These can be delivered inside and outside of the group chat.
- IsComposing indications
  This allows a user in a Group Chat conversation to see when another user is typing a new message/reaction.
- Local Black List
  The terminal/client may support a locally stored Black List to handle incoming Group Chat requests. Users are allowed to qualify undesired incoming Group Chat requests as spam. This prevents subsequent messages from those originators to be shown or even notified to the user. Also, this undesired traffic will not be acknowledged to have been read. The Black List behaviour applies not only to Group Chat but also to Chat and to File Transfer.
- Local Conversation History
  The terminal/client supports a locally stored conversation.
- A Common Message Store
  A Common Message Store for the chat sessions may be used to synchronize the messages between devices. It also allows the user to keep a back-up of important conversations in the network.
  In the device, alignment is expected between the local Conversation History and the synchronization with the Message Store Server.

- User Alias (Display Name)
  A user defined display name can be sent when initiating a communication with another user.
- A long lived Group Chat remains available to its participants
  Once a user initiates a Group Chat, any remaining participant can restart it, even if the Group Chat had been torn down by the Messaging Server because of inactivity.
- Basic Store and Forward feature
  Messages missed because of connectivity issues after joining a Group Chat are stored in the Messaging Server and delivered when the participant rejoins the Group Chat.
- Full Store and Forward feature
  Messages missed because a participant has not yet joined the Group Chat or who was offline when the Group Chat started are stored and delivered when the participant becomes available or joins.
- Closed Group Chat
  A user initiating a Group Chat or Messaging Server can specify that a Group Chat shall be closed, meaning that no one is permitted to add participants to the Group Chat.
- Leaving a Group Chat
  For a Closed Group Chat, a user who explicitly leaves cannot rejoin. For a regular Group Chat, once that Group Chat terminates because of inactivity, a participant who explicitly left cannot rejoin or restart unless he is added by another participant, since that user is no longer on the latest participant list.
  If the Group Chat session is still ongoing, the user may rejoin even after he explicitly left.
- Flexibility to allow multimedia messages within a chat conversation
  Multimedia message exchange is supported in a chat session. However, whether or not multimedia messages are allowed during a Chat session is up to a Service Provider and controlled by a configuration parameter.

A Group Chat can only be initiated by a user belonging to a Service Provider which has deployed a Messaging Server.

A Service Provider may disable the whole Group Chat functionality via the GROUP CHAT AUTH configuration parameter (see Table 85 in Annex A). In case Group Chat is disabled, the client shall not be able to either initiate or participate in a group chat session and reject any group chat session invitation that it receives.

A Closed Group Chat can be set up meaning no new participants may be added. If a Group Chat is closed, the Messaging Server indicates to all group participants that it is closed. A Service Provider may offer the choice to the user, or may only offer either one type of Group Chat or the other.

Once a participant explicitly leaves a Closed Group Chat by sending a SIP BYE request, it is not possible to rejoin since by definition it is not possible to add participants to a Closed Group Chat.

When a Service Provider has deployed a Messaging Server the OMA SIMPLE IM configuration parameter CONF-FCTY-URI (see Table 85 in Annex A) should be correctly set. The CONF-FCTY-URI is used by the device for initiating a normal ad-hoc Group Chat.

It is optional for a Service Provider to provide the Group Chat functionality, so from the terminal perspective, if there is no CONF-FCTY-URI configured, the terminal should not allow the user to add additional parties to the a 1-to-1 chat or to start a Group Chat.

If starting this Group Chat would have increased the number of concurrent chat sessions above the Service Provider configured maximum limit (see MAX CONCURRENT SESSIONS in Annex A), the device would close one of the other active chat sessions (for

example, the chat that has not been used for the longest period of time) before initiating this new one.

There are two types of Store and Forward features for Group Chat:

1. Basic Store and Forward feature, where storage only occurs for a participant who has already joined the Group Chat and that participant has connectivity problems, and
2. Full Store and Forward feature, where messages are stored for a participant if he joins late, or never joins the group chat while it is ongoing, as well as if that participant has connectivity problems.

The Store and Forward feature for Group Chat is only available for a participant when his Service Provider deploys a Messaging Server.

A Group Chat may have participants that have no Store and Forward feature, Basic Store and Forward or Full Store and Forward.

Stored messages are delivered to the participant as follows:

1. When the user sends a request to rejoin the Group Chat. The Messaging Server will deliver the stored messages whether or not the Group Chat is still ongoing.
2. When the user accepts an invitation to join a Group Chat for which there are stored messages.

   NOTE: If the user rejects a Group Chat invitation, all stored messages for that Group Chat are deleted.

3. Once the Group Chat is over or inactive and the Messaging Server knows the user is registered in IMS, or once the Messaging Server Participating Function receives an indication that the user has registered in IMS, the Messaging Server sends an invitation to the participant to deliver the Group Chat messages. If this invitation is not answered then further delivery attempts will be made based on local Service Provider policy. If none of the delivery attempts succeeds and the messages are not delivered, they will expire after an amount of time configured by the Service Provider. If the user rejects the invitation, all stored messages for that Group Chat are deleted.

3.4.2 Interaction with other RCS features

3.4.2.1 Interaction between Full and Basic Store and Forward Group Chat

While a device may use the service capability defined for Full Store and Forward Group Chat to determine who to invite to a Group Chat, a device is still expected to accept incoming Group Chat invitations from users who might not have the Full Store and Forward feature.

A Group Chat hosted by a Messaging Server with the Basic Store and Forward feature could have participants served by another Messaging Server and they could have the Full Store and Forward feature.

3.4.2.2 Interaction in a Group Chat between participants with Store and Forward and participants without Store and Forward

There is nothing to prevent a participant from being part of a Group Chat and be served by a Messaging Server without the Store and Forward feature. Only participants who send messages in the Group Chat will realize a participant is not receiving all messages because they will not receive delivery notifications for their sent messages.

3.4.2.3 Interaction with other RCS features

Interaction of Group Chat with other RCS features is described in section 3.3.2.

If the user wishes to transfer a file to Group Chat participants, this can be done using the procedure in section 3.5.4.2 when supported by the conference focus. Otherwise the user's device must do this by sending the file one by one to each Group Chat participant, and it may or may not appear in the Group Chat window.

3.4.3 High Level Requirements

The following list of high level requirements applies to Group Chat:

- Client devices:

3-4-1 "Delivered" notification request and response

3-4-2 "Displayed" notification request and response
Note that the client device should allow the user to enable or disable the displayed notifications request and response

3-4-3 Delivery of notifications (delivered and displayed) outside a session

3-4-4 IsComposing indications

3-4-5 Procedures associated to the store and forward of both messages and notifications performed by the Messaging Server

3-4-6 Ability to request a regular Group Chat or a closed Group Chat

3-4-7 Ability, based on operator configuration in the device, to indicate support for full Store and Forward Group Chat

- Messaging Server: In addition to the above requirements:

3-4-8 The messaging server may provide interworking of Group Chat to SMS/MMS

3-4-9 The messaging server may provide store and forward of both messages and notifications

3-4-10 This is a function which is provided on the terminating MNO network however a Messaging Server may additionally provide originating store and forward functionality to avoid dependencies with other MNO network implementations.

3-4-11 Both Basic and Full Store and Forward features may be supported

3-4-12 The messaging server may provide the ability to set up a regular Group Chat or a closed Group Chat and shall indicate to all the participants whether the Group Chat is closed or not

3-4-13 The messaging server may store the participant list to be able to recreate the Group chat when it has been timed out by inactivity.

3.4.4 Technical Realization

Group Chat technical realization is based on the "Ad-Hoc Session Mode messaging" as described in [RCS5-SIMPLEIM-ENDORS] and in [RCS5-CPM-CONVFUNC-ENDORS], (depending on the setting for CHAT MESSAGING TECHNOLOGY defined in Table 85 in Annex A).

Support for delivery and display notifications within a Group Chat is added to the functionality endorsed in [RCS5-SIMPLEIM-ENDORS] and [RCS5-CPM-CONVFUNC-ENDORS]. For OMA CPM, also the functionality to support sending "IsComposing" messages within a Group Chat is added.

The Closed Group Chat feature is provided by a Messaging Server, and all participants in such a Group Chat are made aware that it is closed.

The Store and Forward feature for Group Chat is provided by each Group Chat participant's own Messaging Server.

The Full Store and Forward feature of Group Chat builds on the Basic Store and Forward feature. The Full Store and Forward IARI feature tag is used as a service capability, but is

neither carried in the IMS registration nor in the SIP INVITE requests/responses related to Group Chat. A participant's device might only allow a user to choose contacts for a Group Chat who support the Full Store and Forward feature. This depends on the setting of configuration parameter GROUP CHAT INVITE ONLY FULL STORE FORWARD defined in section A.1.3.

To prevent revealing the user identity when transmitted over unprotected links, the client should set the CPIM To header of a Message exchanged in a Group Chat to *sip:anonymous@anonymous.invalid*. For Delivery and Display notifications it will be set as described in section 3.4.4.1.5. As the CPIM From header is needed to identify the sender of the message the user's identity will be provided there and include the display name.

The originating Messaging Server shall always set the CPIM DateTime header in the chat messages it receives. The originating Messaging Server shall also set the CPIM DateTime header and IMDN DateTime element in notifications. In both cases, the Messaging Server shall overwrite any DateTime information provided by the client. A client receiving these requests should therefore rely on these headers containing the correct time rather than on locally available time information.

To allow a user to rejoin or restart a Group Chat, the user's client is required to know the actual focus Session Identity created by the Messaging Server when the Group Chat was initiated. The recommended approach is that any SIP proxy (e.g., Interworking Call Session Control Function [I-CSCF], P-CSCF, IMS-ALG, Session Border Controller [SBC], NAT) in the path between the Messaging Server and the client transparently forwards a received Contact header field from the network being sent towards the UE when the Contact header field contains the "isfocus" feature tag. This is as specified in sections 5.2.7.2, 5.2.7.3, 5.7.5.1 and 5.10.5, of [3GPP TS 24.229].

In normal circumstances for a given RCS Group Chat ID there is at most only a single session is active at a time for an RCS user. In the OMA CPM realization, as detailed in [RCS5-CPM-CONVFUNC-ENDORS], only one Group Chat is supported within a given CPM Conversation.

3.4.4.1 Technical Realization of Group Chat with Delivery and Display Notifications

3.4.4.1.1 Initiating a Group Chat

User A initiates a chat by selecting some of his contacts (Users B, C and so on, up to a limit set by the OMA SIMPLE IM parameter MAX_AD-HOC_GROUP_SIZE – see Annex A) from the address book or from the Chat application in his device, or from the Contact List from the Broadband Access PC client. This choice may be offered only among the contacts known by his devices to be RCS users with Chat capability. It may be offered for all contacts if a Chat interworking service to SMS/MMS is available from the Service Provider (See configuration parameter IM CAP NON RCS, in Table 85 in Annex A).

If the IM CAP NON RCS is disabled, then the device recognizes whether the Chat/Group Chat service is available for a particular contact by using the service capability exchange via Presence or OPTIONS as described in of section 2.6.

When initiating a Group Chat User A may first provide a subject for the conversation that will be provided to all invitees of the group chat. To provide a good user experience, it is recommended to provide it even to those that are invited later when the chat is ongoing. When User A presses the "Send" button, device A initiates a Chat session with the Messaging Server by sending a SIP INVITE request to the conference factory (carrying the subject in the Subject header if one was provided) with a new RCS Group Chat ID. The Messaging Server sends SIP INVITE requests to the other participant users indicated in a recipient-list body in the INVITE request received from User A. The list of invited participants is sent in the Group Chat invitation, and is also sent out to all invited participants.

When a user's client receives a Group Chat invitation from the Messaging Server, the user may accept or reject the invitation. Alternatively the user's client may auto-accept[26] a Group Chat invitation, depending on a configuration parameter IM SESSION AUTO ACCEPT GROUP CHAT as defined in Annex A. A client on which Group Chat has been disabled via the GROUP CHAT AUTH configuration parameter (see section A.1.3.3) shall not notify the user and automatically reject any received SIP INVITE request for a Group Chat with a SIP 488 NOT ACCEPTABLE HERE response.

When at least one invited participant accepts the invitation, the 200 OK response is sent back to User A and the Group Chat is set up. At that moment User A (or the user that accepted) can write a first message in the chat. Any messages sent to the focus during this startup phase before a final response is received from each invited participant are temporarily queued in the Messaging Server Controlling Function until there is a final response from all invited participants. Depending on each participant's response:

- A 200 OK is received, the focus will send all the messages temporarily queued;

When all the final responses have been received, the focus will stop queuing messages.

After acceptance the client shall subscribe to the conference event package to retrieve the list and status of the users in the Group Chat.

User A's device shall also subscribe to the conference event package. The SUBSCRIBE request shall be routed to the Messaging Server Participating Function which shall either proxy the request onwards to the Controlling Function or handle the subscription as a Back-to-Back User Agent (B2BUA), see also section 3.4.4.3.1. When the client receives the resulting SIP NOTIFY requests carrying the conference state information, the identity of each user shall be matched against the Contact List in the device to present a user friendly name. If a user is not found in the Contact List, the display name provided in the conference state, if any, for that user should be used. As a complement to [RCS5-SIMPLEIM-ENDORS] and [RCS5-CPM-CONVFUNC-ENDORS], the notifications pertaining to the conference event package should convey information about the pending participants (i.e. the "*pending*" state is used in the <*status*> element of the corresponding endpoint as per [RFC4575]).

The Messaging Server will open sessions to Users A, B, C and so on, up to a configured limit which should be set to the same OMA SIMPLE IM parameter value configured in the clients, i.e. MAX_AD-HOC_GROUP_SIZE.

In the user interfaces of the receivers' client on which the GROUP CHAT AUTH configuration parameter is enabled (see section A.1.3.3), a notification (UI dependent) shall be displayed to inform the user about the incoming invitation. This notification should clearly state that it is an invitation to a Group Chat making the users aware of this fact and should take into account the IM SESSION AUTO ACCEPT GROUP CHAT configuration parameter defined in section A.1.3.3. This notification can also indicate the other users that were invited as well as the subject of the conversation if one was provided.

The supported content types in the SDP exchanged in the SIP INVITE request and the associated 200 OK response shall be indicated as follows:

- In the SDP of the SIP INVITE request and response, the a=accept-types attribute shall include only *message/cpim*, i.e., "*a=accept-types:message/cpim*".
- If multimedia content within a Chat session is a requirement the configuration parameter MULTIMEDIA IN CHAT in Annex A section A.1.3.3 is set to enabled. Therefore in the SDP of the SIP INVITE request and response, the *a=accept-wrapped-types* attribute

[26] Note that the Service Provider multidevice policy has to be consistent with the Group Chat auto-acceptance policy.

shall include either *, or a complete list of all content types supported during the Chat session (including at least *text/plain*, *message/imdn+xml* and *application/im-iscomposing+xml*), e.g., *a=accept-wrapped-types:**.

- If multimedia content within a Chat session is not a requirement, the configuration parameter MULTIMEDIA IN CHAT in Annex A is set to disabled. Therefore in the SDP of the SIP INVITE request and response, the *a=accept-wrapped-types* attribute shall only include *text/plain*, *message/imdn+xml* and *application/im-iscomposing+xml.* If File Transfer using HTTP is supported (see section 3.5.4.8) then the a=accept-wrapped-types attribute shall also include *application/vnd.gsma.rcs-ft-http+xml.* If Geolocation PUSH is supported (see section 3.10.4.1.3), then the *a=accept-wrapped-types* attribute shall also include *application/vnd.gsma.rcspushlocation+xml*. To transfer multimedia content during a chat, File Transfer is used.

  NOTE: During session setup the client and the conference focus shall also take into account the procedures described in section 3.5.4.2 and when they support File Transfer via HTTP in the Group Chat also those in section 3.5.4.8.1. When they support Geolocation PUSH in the Group Chat they shall also take into account the procedures in section 3.10.4.1.3.

Once the Group Chat has been initiated, the focus Session Identity uniquely identifying that Group Chat may be kept by the original initiator's Messaging Server Controlling Function either for an amount of time configurable by the Service Provider, or until the original initiator's Messaging Server Controlling Function no longer authorizes other users to restart this Group Chat (e.g. because of a Service Provider policy when the initiator explicitly left the chat). Depending upon service provider policies, this focus Session Identity can then be used by any of the participants in the Group Chat to restart it by simply attempting to rejoin the Group Chat as described in section 3.4.4.1.7.

3.4.4.1.2 Adding participants to a Group Chat

Once a Group Chat is established, the local Service Provider policy decides whether only the initiator is allowed to add participants to the Group Chat or whether any participant is allowed to add more participants. A Service Provider may choose to have a local policy that allows participants that are their own subscribers to add participants, but participants from other Service Providers would not be allowed.

The maximum number of participants allowed and the current user count for a running group chat is notified by the focus in the maximum-user-count and user-count elements as defined in [RFC4575] when the client subscribes to the conference event package. Participants may be added provided the maximum-user-count is not reached or the focus's Service Provider policy allows it. If these values are not present in the conference event package then that the MAX_AD-HOC_GROUP_SIZE configuration parameter may be used instead.

Participants on the participant list can be added again (re-invited), provided that they are not in the active or pending state or have left the chat involuntarily. This way participants that have not accepted the INVITE request before it has timed out, or participants that left the chat voluntarily (see section 3.4.4.1.3.1) can be added again.

For a Closed Group Chat (see section 3.4.4.2), no one can add participants. If the Group Chat is a Closed Group Chat, the Messaging Server will return an error response to the SIP REFER request. If adding participants fails because of one of the reasons above, it is expected that the Messaging Server's error response include a Warning header and appropriate explanatory text as per the [RCS5-SIMPLEIM-ENDORS] or [RCS5-CPM-CONVFUNC-ENDORS] (depending on the setting for CHAT MESSAGING TECHNOLOGY see Table 85 in Annex A).

When adding participants, as a clarification to [RCS5-SIMPLEIM-ENDORS] or [RCS5-CPM-CONVFUNC-ENDORS] (depending on the setting for CHAT MESSAGING TECHNOLOGY see Table 85 in Annex A), the client shall :

- include an RCS Group Chat ID in the REFER request set to the RCS Group Chat ID of the pertaining Group Chat,

include a Subject header in the REFER request set to the pertaining Group Chat subject, if the pertaining Group Chat was created with a subject.

3.4.4.1.3 Closing Group Chat

A SIP/MSRP session established between one of the user's devices and the network allowing a user to participate in a Group Chat will be closed for either one of the following reasons:

- The user explicitly indicates that he's not willing to take part in the Group Chat anymore (see section 3.4.4.1.3.1)
- An error condition occurs preventing a particular device from further maintaining an active session that allows a user to participate in a Group Chat (see section 3.4.4.1.3.2)
- Based on local service provider policies, the Controlling Function no longer wishes to maintain the active sessions of an ongoing Group Chat and therefore forces the closing of the remaining active sessions (see section 3.4.4.1.3.3)

3.4.4.1.3.1 Explicit Departure

Any of the participants can voluntarily leave the Chat. When leaving voluntarily, a conversation history will exist in the user's device history with the messages associated with the chat up to the point the user left.

When a participant indicates their desire to voluntarily leave the Group Chat session their device shall send a SIP BYE request that includes a *Reason* Header field (as defined in [RFC3326]) with the protocol set to *SIP* and the protocol-cause set to *200* (e.g. SIP;cause=200;text="Call completed"). In addition, the client shall unsubscribe from receiving the Chat participant information. When it receives a SIP BYE request the Controlling Function will convey with a new conference state event package notification to the remaining participants and in case of voluntary departure, remove the user from the participant list. In this new conference state event package notification the Controlling Function shall include additional elements and values defined in [RFC4575] to indicate why the participant is not taking part in the conversation any longer. If the participant left voluntarily (that is, the SIP BYE request included a *Reason* header field with the protocol set to *SIP* and the protocol-cause set to *200*), the conference focus shall indicate the departed participant's status as "disconnected" and include a disconnection-method element the value of which shall be set to "departed".

When the chat is (re)started, any participant can also leave or decline the Group Chat by rejecting the Group Chat invitation with a 603 Decline response. When receiving a SIP 603 DECLINE response to a SIP INVITE request for a Group Chat, the Controlling Function will remove the participant that declined from the participant list and convey a new conference state event package notification to the remaining participants. In this new conference state event package notification, the controlling function shall set the departed participant's status to "disconnected" and include the disconnection-method element set to a value of "failed" with a reason sub-element set to code *603* *<reason>SIP;cause=603;text="Declined"</reason>*.

NOTE: When the user explicitly leaves a regular Group Chat (i.e. as opposed to a Closed Group Chat as defined in section 3.4.4.2), their client may store the Group Chat's IM Session identity for some time to offer the user the option to rejoin. If the user makes use of this option the device shall handle this

according to the procedures described in [RCS5-SIMPLEIM-ENDORS] or [RCS5-CPM-CONVFUNC-ENDORS]. Once that Group Chat terminates because of inactivity, that participant who explicitly left cannot rejoin or restart unless he is added by another participant, since that user is no longer on the latest participant list. When an attempt to rejoin results in a SIP 404 Not Found response, the chat should be considered to be no longer active.

When User C leaves the Group Chat voluntarily, the other users that have subscribed to the conference focus will be notified in the chat through a predefined indication “User C has left the conversation,” and their devices will remove him from the displayed participants.

An RCS client receiving a notification of a participant leaving the Group Chat voluntarily, either by closing (i.e. using a SIP BYE request that includes a *Reason* header field with the protocol set to *SIP* and the protocol-cause set to *200*) or rejecting the Group Chat session invitation with a SIP 603 DECLINE response shall remove the participant from the locally stored participant list associated with the Group Chat.

A participant voluntarily leaving the Chat is removed from the Group Chat participant list used for restarting the chat as specified in section 3.4.4.1.7 by both the Controlling Function and the clients receiving a notification indicating this voluntary departure.

3.4.4.1.3.2 Involuntary departure

A user can also leave the session involuntarily (e.g. due to loss of connectivity or other error situations). In this case, if it is still capable of doing so, the client or otherwise, the network element detecting the error situation should generate a SIP BYE request that includes a *Reason* Header field with the protocol-cause set to a value other than *200* (i.e. the protocol-cause must not be the value used for voluntary departure in section 3.4.4.1.3.1). It is recommended to use a *Reason* header field with the protocol set to *SIP* and the protocol-cause set to *503* (e.g. SIP;cause=503;text=”Service Unavailable”) as was defined in [3GPP TS 24.229] for bearer loss detected by the P-CSCF also for other network elements.

When a SIP BYE request carrying a *Reason* header field with a protocol-cause other than *200* is received by the Controlling Function, the Controlling Function shall send SIP NOTIFY requests to the remaining participants of the chat indicating in the Conference State event package notifications that the user is no longer part of the session. The Controlling Function shall represent the fact that the departure was involuntary by setting the participant’s status to “disconnected” and including a disconnection-method element of which the value shall be set to “booted”.

NOTE1: As it cannot be guaranteed in general that all network elements detecting an error situation will include a Reason header field nor that there are no legacy clients connecting to the network that do not include a Reason header field, it is recommended that the Messaging Server allows for a Service Provider policy controlling whether a received SIP BYE request without Reason header field is handled as a voluntary or involuntary departure. If such a policy is not provided, it is left to implementation.

NOTE2: As described in section 3.4.4.3.1, depending on whether store and forward is supported the Participating Function may not always relay a SIP BYE request indicating involuntary departure that is initiated by a client or another network element located on the path between the Participating Function and the client. If store and forward is supported, the Participating Function will not forward the SIP BYE to the Controlling Function. The disconnection is transparent to Controlling Function and the participants.

3.4.4.1.3.3 Closing of the Group Chat

The remaining sessions for a Group Chat are closed by the Controlling Function amongst others when in following cases:

1. less than the minimum number of active participants as defined in the Messaging Server, for a Group Chat remain in the Group Chat, or
2. when a chat inactivity timeout expires, or
3. based on local policy in the Messaging Server, e.g. if the originator leaves the Group Chat.

   NOTE: The active participants are the ones in “connected” state or in the “pending” state (i.e. the ones from which a final response has not yet been received).

When closing these sessions, a conference focus may decide based on service provider policy to keep the focus Session Identity. This may be for example done in following cases:

- Case 1 above, when there are users on the participant list that have left involuntarily (see section 3.4.4.1.3.2)
- Case 2 above

When it decides to keep the focus Session Identity and associated information, the conference focus shall maintain the information for at least one month.

NOTE: It is recommended that a Participating Function providing store and forward functionality for Group Chats as described in sections 3.4.4.3 and 3.4.4.4 stores deferred Group Chat messages for at most one month to ensure that the original focus can be used should there be a need to restart the session as a consequence of the forwarding of the deferred messages.

When the Messaging Server no longer keeps the focus Session Identity for the Group Chat, any future attempt by a user to join or restart the Group Chat identified by the focus Session Identity will fail. The Controlling Function shall indicate this by including a *Reason* header field with the protocol set to *SIP* and the protocol-cause set to *410* (e.g. SIP;cause=410;text=”Gone”) in the SIP BYE request that it sends to the remaining participants. A client receiving a SIP BYE request including a *Reason* header field with the protocol set to *SIP* and the protocol-cause set to *410* may take this into account when restarting the Group Chat (see section 3.4.4.1.7) and avoid sending a rejoin request to the focus Session Identity.

Otherwise the Messaging Server keeps the focus Session Identity. This allows any of the participants that still were in the Group Chat when the session was closed to send a rejoin request that will result in the Group Chat being restarted. In that case, the Messaging Server also keeps the type of Group Chat (e.g. Closed) and the list of participants that were present when the Group Chat was torn down, and uses that list to check who is authorized to restart the Group Chat, and then to invite those participants when it receives the rejoin request from an authorized participant as described in section 3.4.4.1.7. When the Session Identity is maintained in case 2 above or in case 1 above for the situation where some of the

participants left the session involuntarily or didn't explicitly accept or reject the invitation, the Controlling Function shall include a *Reason* header field with the protocol set to *SIP* and the protocol-cause set to *480* (e.g. SIP;cause=480;text="Bearer unavailable") in the SIP BYE request that it sends to the remaining participants. In other scenarios where the messaging server keeps the focus Session Identity, it shall include a *Reason* header field, but that may relay different values.

Even if the focus indicated that it keeps the Session Identity and the participant list (i.e. the BYE request received by the clients included a Reason header with the protocol set to SIP and the protocol-cause set to a value different than 410), the clients participating in a Group Chat shall also keep the latest participant list. If a rejoin to an inactive Group Chat fails, the client can then restart the Group Chat as a new Group Chat using this latest participant list from the last Group Chat as described in section 3.4.4.1.7.

To avoid that users are removed from an ongoing session, the idle time during a Group Chat should be monitored by the Controlling Function ensuring that the session is torn down for all users at the same time. Unlike the situation for a 1-to-1 Chat where the Participating Function may tear down the session also in normal circumstances, any idle session monitoring on the Participating Function, if provided, shall therefore use timeouts that are significantly larger than the timers used in the controlling function. To allow configuring these Participating Function timers for intervening in error situations (e.g. loss of connectivity between the Participating and the Controlling Function), the maximum allowed idle time on the Controlling Function shall not be larger than 300 sec (i.e. 5 min). Because they are not relayed to all Participating Functions involved in the Chat, the Controlling Function shall not take the messages relayed over MSRP for delivering disposition notifications into account when monitoring the idle time.

NOTE: Whether to provide idle time monitoring of a Group Chat in the participating function for intervening in error situations is an implementation decision.

3.4.4.1.4 Chat message size limitations

This maximum size is controlled through the MaxSize1ToM configuration parameter defined in Annex A. Endpoints and the Messaging Server are expected to make use of the SDP attribute a=max-size to indicate the maximum message size to participants.

If the user attempts to send a message larger than this limit, the message is not sent, and the user should be informed that messages of that size cannot be sent in the conversation.

3.4.4.1.5 Delivery and Display notifications within Group Chat

Each message sent within a Group Chat may request a delivery notification and may request a display notification, similar to the previously described 1-to-1 chat (see section 3.3.4.1).

The recipient client generates the delivery or display notifications as described for 1-to-1 chat (see section 3.3.4.1), with the difference that the CPIM *To* header shall be set to the identity of the sender of the message, found in the CPIM *From* header of the incoming message, instead of to an anonymous URI. The identity to be used from the CPIM *From* header could be a public gruu for the sender, a URI for the sender plus the sip.instance value for the sender, or it could simply be a URI for the sender. For examples of how a public gruu or sip.instance value is carried in a CPIM *From* header, see section 3.4.4.1.8. While the examples in section 3.4.4.1.8 address the CPIM *From* header case, they also apply for the value to be used in the CPIM *To* header.

This requires that the Messaging Server support Private Messages within Group Chat.

If there is no on-going Group Chat in which to send these notifications, they shall be sent using SIP MESSAGE. Note that the SIP MESSAGE may not arrive at the sending device in

a multidevice scenario if a device identifier (i.e. public gruu or sip.instance value as per section 2.11.3) is not included in the SIP MESSAGE request.

3.4.4.1.6 Interworking to SMS/MMS

For a Group Chat the behaviour for interworking to SMS/MMS is similar to interworking of a 1-to-1 Chat described in section 3.3.4.1.6 with the same entities being involved. The only difference being that the decision to interwork may be taken by the Controlling Function of the Messaging Server based on the same criteria used by the Participating Function for the case of a 1-to-1 session (e.g., based upon error), or by the terminating Participating Function (e.g. based upon error or Service Provider policy). Furthermore as described in [RCS5-CPM-IW-ENDORS] and [RCS5-3GPP-SMSIW-ENDORS] the IWF will subscribe to the participant information and use that information to inform the SMS or MMS user of who also is taking part in the Group Chat.

Note that messages sent during any interval that a participant becomes unavailable or loses connectivity during the Group Chat will only be stored for that participant if either the basic or full Store and Forward feature is available for that participant. See sections 3.4.4.3 and 3.4.4.4 for more information on the Group Chat Store and Forward features.

3.4.4.1.7 Restarting a Group Chat

When a Group Chat has been closed due to inactivity, it may be restarted at any time by any of the participants. In order to do so, the RCS client will try to rejoin using the focus Session Identity and same RCS Group Chat ID of the previous Group Chat session. Depending on Service Provider policies, the Group Chat may be automatically restarted as explained below or a SIP 404 error response will be returned. If a SIP 404 error response is returned, the RCS client shall initiate a new Group Chat as per section 3.4.4.1.1 re-using the same RCS Group Chat ID and with latest participant list it has available for the Group Chat.

NOTE1: No specific handling for Group Chat restarts is required for non-2xx SIP responses other than those mentioned in this section.

The participant list should include the participants in the pending state. A client shall not restart an already established Group Chat. For OMA CPM, the RCS client shall restart the Group Chat with the same Contribution-ID and the same Group Chat ID (Conversation-ID) of the previous Group Chat session when restarting a Group Chat as specified in [RCS5-CPM-CONVFUNC-ENDORS].

NOTE2: Optionally, based on Service Provider policies, the Participating Function may return a SIP 404 error response to the rejoin request, which will as described above result in an initiation of a new Group Chat with a new Session Identity, re-using the same RCS Group Chat ID, and the latest participant list. From client perspective, the same procedure described in this section therefore applies.

If, when using that last method, the number of participants in the participant list is larger than the maximum number of participants allowed (i.e., as configured using the MAX_AD-HOC_GROUP_SIZE configuration parameter) the user should be prompted to reduce the number of participants before initiating the Group Chat and a new RCS Group Chat ID is assigned.

When the Messaging Server Controlling Function receives an incoming SIP INVITE request with a focus Session Identity for a Group Chat that is not in progress, it checks whether it still has the focus Session Identity along with the last participant list. If so, it checks whether the user is authorized to restart the Group Chat identified by the focus Session Identity in

the SIP INVITE request, and if so, restarts the Group Chat using the associated participant list and as a Closed or regular Group Chat depending on the type it was before.

If the Messaging Server Controlling Function no longer recognizes the focus Session Identity, it returns a SIP 404 error response and if the focus Session Identity is available but the user is not authorized to restart the Group Chat, it returns a SIP 403 error response. In both cases, the user's device initiates a new Group Chat as per section 3.4.4.1.1, using the last participant list it had already stored to build the URI-list in the SIP INVITE request

It may happen that more than one participant in a Group Chat that was closed because of inactivity will restart the Group Chat at the same time, resulting in two or more conference foci being allocated using the same RCS Group Chat ID (i.e. the OMA SIMPLE IM Contribution-ID or OMA CPM Conversation-ID). In that case and since rejecting a Group Chat invitation or terminating an ongoing Group Chat session with a SIP BYE request implies removing the participant from the Group Chat, an RCS client receiving such an invitation shall behave as follows to keep one of the Group Chats and allow all the others to become idle:

- If more than one Group Chat invitation is received with the same RCS Group Chat ID, the RCS client shall accept or reject all the invitations according to the normal procedures depending whether the user wants to rejoin or not.
- If a Group Chat invitation is received with the same RCS Group Chat ID of an already established Group Chat, the RCS device will auto accept the new Group Chat session. The participant list contained in the SIP INVITE request has to be compared with the local participant list and if one or more participants are found in the local list and not present in the incoming SIP INVITE request, the RCS client will automatically add those participants to the new Group Chat session as per section 3.4.4.1.2.
- The RCS client shall be able to receive all incoming message from any of the established Group Chat sessions with the same RCS Group Chat ID.
- The RCS client will send messages to the Group Chat using only the latest established Group Chat session with the same RCS Group Chat ID. This will allow the rest of Group Chat sessions to time out due to inactivity.
- If the participant explicitly leaves the Group Chat, all the Group Chat sessions with the same RCS Group Chat ID will be terminated by the RCS client by sending a SIP BYE request.

3.4.4.1.7.1 Race conditions

Since a Group Chat can be restarted by two participants simultaneously, race-conditions exist between rejoin requests coming from the client and SIP INVITE request originated by the Controlling Function. As the middle element, most of these situations will be detected by the Messaging Server Participating Function that shall handle these situations as follows:

1. For the case where the Messaging Server Participating Function receives an incoming SIP INVITE request from the client for a Group Chat for which a SIP INVITE request was already sent (matching shall be done based on the focus Session Identity or the RCS Group Chat ID) (i.e. the INVITE requests have crossed between the client and the Participating Function):
   a) If no session is established yet with the Controlling Function (see also sections 3.4.4.3 and 3.4.4.4), the Messaging Server Participating Function shall forward this INVITE request from the client to the conference focus and handle the SIP INVITE request from the client as a regular B2BUA with this session setup to the Controlling Function.
   b) Otherwise the Messaging Server Participating Function shall accept the SIP INVITE request from the client, establish the MSRP channel and forward any messages and

notifications received from the client in the already established MSRP session with the Controlling Function.

Messages and notifications received from the Controlling Function shall only be forwarded in the MSRP channel to the client that was last to be established. Based on local policy, the Messaging Server Participating Function shall terminate the unused session by sending in the corresponding SIP dialog a SIP BYE request carrying a Reason header field with the protocol set to SIP and the protocol_cause set to 480 (e.g. sip;cause=480;text="bearer unavailable").

This means, that in all cases where such a race condition occurs temporarily, two sessions are established between the client and the Participating Function and only one between the Controlling Function and the Participating Function. Between the client and Participating Function only the MSRP session that was last to be established shall be used.

2. For the case where the Messaging Server Participating Function receives an incoming SIP INVITE request from the Controlling Function for a Group Chat for which a SIP INVITE request was already sent (matching shall be done based on the focus Session Identity or the RCS Group Chat ID) to the Controlling Function (i.e. the INVITE requests have crossed between the Controlling Function and the Participating Function), the Messaging Server Participating Function may either
   a) forward this INVITE request to the client or
   b) accept both the session from the Controlling Function and the rejoin request from the client and link both dialogs as a B2BUA. In that case when the Controlling Function accepts the INVITE request, the Participating Function shall establish the MSRP session and only use the last MSRP channel to be established until either the Controlling Function closes one of the sessions, closes the entire chat or the user leaves the Chat. In the last case the Messaging Server Participating Function shall send the corresponding SIP BYE request in both sessions.

Also the Controlling Function shall accept a rejoin request received from a participant for which there was an outstanding INVITE request. It shall ensure that only one session is used and that messages from the participant are not returned in the other session. To achieve this, it is recommended to only send messages in the last MSRP channel to be established. Once it has received messages or notifications over that connection it may close the other session by sending in the corresponding SIP dialog a SIP BYE request carrying a Reason header field with the protocol set to SIP and the protocol_cause set to 480 (e.g. sip;cause=480;text="bearer unavailable").

3.4.4.1.8 Multidevice handling and the Common Message Store

Multidevice handling occurs when a user has more than one device (e.g., PC and mobile). When a new Group Chat is initiated and an invited user, User B, has multiple devices registered at the same time, the Group Chat session request shall be forked to each device for that user by the terminating Messaging Server Participating Function. Forking at the Messaging Server is further described in 3.4.4.1.8.1.

When a user wishes to restart a chat on a different device, it may not have the latest information on the Group Chat. In particular it will not have the focus Session Identity for the Group Chat if it did not receive the initial SIP INVITE request. This would occur if the device had not been registered in IMS when the initial INVITE request was sent out.

When a Common Message Store is available for the user, the Group Chat messages are stored in the Common Message Store as they pass through the user's Participating Function. If a network initiated SIP BYE request is received, then in addition to storing the Conversation-ID (if present) and Contribution-ID, the focus Session Identity for the Group Chat, whether it was closed or regular, and a message containing the current list of

participants need to be stored. This implies that the Messaging Server Participating Function is required to keep track of the list of participants for the Group Chat by subscribing to the participant information and keeping the relevant information received in the corresponding SIP NOTIFY requests. The messages are synchronized with the Message Store Server as specified in [RCS5-CPM-MSGSTOR-ENDORS].

When chat messages are sent within a Group Chat, the device identifier is required to be set by the sender in the CPIM *From* header in the MSRP SEND request. To do this, the CPIM *From* header in messages should carry the device identifier, which is either a public gruu or a sip.instance value as defined in section 2.11.3.

As examples:

- For a public GRUU, the CPIM From header would be set as follows: *From: <sip:mysipuri@example.com;gr=hdg7777ad7aflzig8sf7>*
- For a sip.instance value, the CPIM From header would be set as follows: *From: <sip:mysipuri@example.com?Accept-Contact=+sip.instance%3D%22%3Curn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6%3E%22%3Brequire%3Bexplicit>*

  NOTE1: URI parameters require escaping in the above example

  NOTE2: the placement of the +sip.instance inside the angle brackets is only valid in the CPIM From header. In particular, the +sip.instance shall never be placed in the Request-URI of a SIP request. If it is needed in a SIP request, it shall be carried in a Contact header if identifying the sender or in an Accept-Contact header if identifying the recipient as defined in section 2.11.3.

When Group Chat messages are stored, the device identifier of the sender is stored with the message. When these messages are delivered, the device identifier for the sender of each message is set in the CPIM From header in the same way it was set by the original sender, as shown above.

3.4.4.1.8.1 Forking on the Messaging Servers

The forking procedures for Group Chat are the same as those described for 1-to-1 chat in section 3.3.4.1.7.1.

Additionally, each device shall subscribe to the conference state event package, once it accepts the INVITE. The Participating Function serving that user shall make sure that there is only one subscription maintained per user towards the Controlling Function.

If a client indicates voluntary departure from the Group Chat when there is no active device as per section 3.4.4.1.3.1, the terminating Participating Function shall close the session towards the other clients and indicate to the Controlling Function that the user has left the group chat.

If a client leaves a Group Chat involuntarily from one of their devices as per section 3.4.4.1.3.2 and there is no user device that has accepted the Group Chat, the Messaging Server shall continue forking incoming messages to the rest of the user devices.

3.4.4.1.9 Group Chat Service Identification when CHAT MESSAGING TECHNOLOGY is CPM

When CHAT MESSAGING TECHNOLOGY (see Table 85 in Annex A) is set to CPM, the RCS client shall populate the P-Preferred-Service header field in all CPM requests with the CPM Feature tag defined for the service, as described in [RCS5-CPM-CONVFUNC-ENDORS]. The S-CSCF or AS that performs the service assertion in the originating network

shall add the P-Asserted-Service header field set to the value of the asserted CPM service ICSI (i.e. “urn:urn-7:3gpp-service.ims.icsi.oma.cpm.session” for CPM chat, or “urn:urn-7:3gpp-service.ims.icsi.oma.cpm.deferred” for deferred delivery done as part of the Store and forward realization) and remove the P-Preferred-Service header field before further routing the request.

A receiving network element and RCS client should ignore any SIP header fields that they do not understand (e.g. P-Preferred-Service, or P-Asserted-Service header fields).

3.4.4.2 Technical Realization of a Closed Group Chat

In order for a device to request that a particular group chat remain closed to the addition of new participants, the device sets the *a=chatroom* attribute defined in [IETF-DRAFT-Chat] in the SDP of the SIP INVITE request with the CPM reserved chat-token value, as described in [RCS5-CPM-CONVFUNC-ENDORS], to indicate the RCS Closed Group Session in the SIP INVITE. The SDP attribute value shall be :
a=chatroom:org.openmobilealliance.groupchat.closed;

If a Service Provider only wishes to provide Group Chats that are closed, then even if the SDP offer did not include the *a=chatroom* attribute, a Closed Group Chat is created and the *a=chatroom* attribute is returned to the sender in the SDP answer in the 200 OK to the Group Chat initiator.

For a Closed Group Chat, the *a=chatroom* attribute (i.e. “*a=chatroom:org.openmobilealliance.groupchat.closed”*) is placed in the SDP offer in the SIP INVITE request sent to invited participants:

A device recognizing this attribute may disable the possibility of adding participants to the Group Chat. A device not recognizing this attribute shall ignore it, but any attempt by the device to add participants will result in an error being returned from the Messaging Server.

3.4.4.3 Technical Realization of Basic Store and Forward functionality for a Group Chat Participant

A Messaging Server may provide the Basic Store and Forward feature for users participating in Group Chats. This feature applies no matter whether this Service Provider or another Service Provider is hosting the Group Chat. This feature applies only to participants who have initiated a Group Chat or have previously accepted a Group Chat invitation and are thus participating in a Group Chat as described in section 3.4.4.1.

3.4.4.3.1 Storing messages and notifications for a participant who loses connectivity

If a Messaging Server implements the Group Chat Store and Forward feature, then if a Messaging Server Participating Function serving a Group Chat participant detects that the participant is unreachable (e.g. loss of network coverage), it becomes an endpoint for the Group Chat. In this case it shall store any messages and disposition notifications (but not isComposing Notifications) received for that participant and keep the dialogs for the Chat session alive by sending timely refresh requests as defined in [RFC4028]. This way, the conference focus continues to keep the participant in the participant list for the Group Chat. Furthermore the Messaging Server Participating Function shall ensure that it has a subscription to the conference state for the duration of the Group Chat, or until the user re-joins. Upon detecting that the participant is unreachable the Messaging Server Participating Function shall either initiate an own subscription to the conference state on behalf of the user after terminating the client’s subscription or only terminate the leg towards the client depending on whether it proxies the subscription requests or acts as a B2BUA (see also section 3.4.4.1.1). A conference focus shall accept subscriptions to the conference event package from Participating Functions that subscribe on behalf of an RCS user.

NOTE1: The Controlling Function cannot be assumed to support multiple subscriptions per participant, if this situation occurs, it may terminate all but the last subscription. In order to come to a predictable behaviour, a Participating Function proxying the client's subscriptions should therefore terminate the client's subscription before initiating its own one and upon a rejoin of the client terminate the own subscription before forwarding the client's subscription.

A Messaging Server knows a participant has lost connectivity if it does not receive a SIP BYE request from that user, but it notices TCP or MSRP connectivity issues with that participant, or receives a SIP BYE request from client side indicating involuntary departure as specified in section 3.4.4.1.3.2 (e.g., containing a Reason header field set to response code 503 (Service Unavailable), as specified in section 5.2.8.1.2 of [3GPP TS 24.229]).

NOTE2: SIP BYE requests indicating voluntary departure (as specified in section 3.4.4.1.3.1) and SIP BYE requests received from the controlling function side shall simply be relayed on the other leg of the B2BUA as for the case where the Group Chat Store and Forward feature is not provided.

3.4.4.3.2 Rejoining a Group Chat after temporary disconnection, Group Chat still ongoing

When a Messaging Server Participating Function detects that one of its users' devices is attempting to rejoin an ongoing Group Chat, it will deliver any stored messages and the notifications pertaining to this user's device, and connect the user's session with the ongoing session that the Messaging Server Participating Function controls on behalf of the user.

The user is made aware of the current list of participants in the Group Chat once he has joined successfully and issues a SUBSCRIBE to the conference state for the Group Chat. The Messaging Server Participating Function shall upon receiving this request either forward it to the Controlling Function and terminate its own subscription or connect the client's subscription as a B2BUA to its own one. All participants with any one of these <status> values: 'connected', 'disconnected' or 'pending' shall be included in the resulting SIP NOTIFY request, not just the ones in 'connected' state.

3.4.4.3.3 Rejoining a Group Chat after temporary disconnection, Group Chat is over

When a Messaging Server Participating Function detects that a user's device is attempting to rejoin an ongoing Group Chat which is no longer active, it will forward the rejoin request to the controlling function and deliver any stored messages and the notifications pertaining to this user's device (see Annex B.1.4), and when no Group Chat session was established during forwarding, terminate the session to the client immediately after delivering the last stored message or notification.

As for the case of a 1-to-1 session described in section 3.3.4.1.4, when stored application messages need to be forwarded this will only be done in case the client that accepted supports the application service as indicated in the *a=accept-wrapped-types* SDP attribute and the Contact header field that the client provided in the SIP INVITE request when rejoining the session. Otherwise, according to the operator local policy the messages related to the application shall either remain stored until another client comes online or be converted by the Messaging Server Participating Function into a format that is supported by the client (e.g. a plain text message carrying a link or a descriptive text of the application message's content).

When the receiving RCS Client sends any messages, explicitly leaves the Chat (as described in section 3.4.4.1.3.1) or adds a new participant (as described in section

3.4.4.1.2), the Messaging Server Participating Function shall ensure that the Group Chat is restarted on behalf of the user using the procedure described in section 3.4.4.1.7 and forward the message or action from the user in this session. For notifications sent by the receiving client, the Messaging Server Participating Function shall either send them in a restarted Group Chat session or as SIP MESSAGE requests. In the latter case, these SIP MESSAGE requests shall carry the same RCS Group Chat ID as the Group Chat so that the receiving user can associate them with the Group Chat instead of to a 1-to-1 Chat. The Messaging Server Participating Function should close the delivery session as soon as all stored messages are delivered.

NOTE: When the client subscribes to the conference state event package and the Group Chat is not active (i.e. there is no session between Participating and Controlling Function), as an optimisation this subscription can be accepted by the Messaging Server Participating Function that will consequently generate a SIP NOTIFY request including the last known participant information. In this case the Messaging Server Participating Function shall have to subscribe to the conference state event package when the Group Chat is restarted and link this subscription with the one from the client as a B2BUA. If the client's SIP SUBSCRIBE request is received in a Group Chat that was restarted, it shall be handled as described in section 3.4.4.1.1.

3.4.4.3.4 Delivering messages and notifications for a participant, Group Chat is over

When the Messaging Server Participating Function detects that a recipient user's device is now available in IMS, stored chat messages and the notifications targeting this device are delivered as described in section 3.3.4.1.4, with following differences (see Annex B.1.17):

- the P-Asserted-Identity and Contact headers shall include the same values of the corresponding headers in the latest received Group Chat invitation,
- the Referred-By header value shall contain the address of the user who initiated the Group Chat and

  NOTE: Whether the Messaging Server Participating Function initiates this action immediately or only after a timeout when it can be assumed that the client will not send an automatic rejoin, is left as an implementation decision. In both cases, the Messaging Server Participating Function should be prepared to handle a race condition between the INVITE request that it sends and a rejoin sent from the client. By delaying the INVITE request, that scenario becomes more unlikely though.

When the receiving RCS Client sends any messages, explicitly leaves the Chat (as described in section 3.4.4.1.3.1) or adds a new participant (as described in section 3.4.4.1.2), the Messaging Server Participating Function shall ensure that the Group Chat is restarted on behalf of the user using the procedure described in section 3.4.4.1.7 and forward the message or action from the user in this session. For notifications sent by the receiving client, the Messaging Server Participating Function shall either send them in a restarted Group Chat session or as SIP MESSAGE requests. In the latter case, these SIP MESSAGE requests shall carry the same RCS Group Chat ID as the Group Chat so that the receiving user can associate them with the Group Chat instead of to a 1-to-1 Chat. The Messaging Server Participating Function shall close the delivery session as soon as all stored messages are delivered. When the user explicitly leaves the session the Messaging Server Participating Function shall next to the forwarding of this action also discard any remaining stored messages.

Forwarding of application messages shall be done as specified in section 3.4.4.3.3. If the first message to be delivered in the Chat is an application message (e.g. a File Transfer via HTTP XML body), the messaging server shall include an Accept-Contact header field in the SIP INVITE request carrying the IARI associated to the application service along with *require* and *explicit* parameters.

3.4.4.3.5 Delivering messages and notifications for a participant, Group Chat is still ongoing

When the Messaging Server Participating Function detects that a recipient user's device is now available in IMS and the session with the Controlling Function is still active, stored chat messages and notifications are delivered as described in section 3.4.4.3.4 with the Messaging Server Participating Function's B2BUA linking the session towards the client with the one that was established with the Controlling Function. The Messaging Server Participating Function shall either forward the subscription to conference state request from the client to the Controlling Function after terminating the own subscription or link this subscription from the client to its own subscription as a B2BUA.

3.4.4.3.6 Active session from Controlling Function deactivated while forwarding messages

In case the Chat Session is terminated from the Controlling Function while the deferred delivery is still ongoing, the session to the client may per Service Provider policy:

- be maintained until all messages and notifications have been delivered or
- be closed and then immediately restarted to push the remaining messages and notifications: the Group Chat session to the Controlling Function may also be restarted as described in section 3.3.4.1.7 to handle content sent by the client.

3.4.4.3.7 Inactive session from Controlling Function activated while forwarding messages

In case the delivery started without an active session from the Controlling Function and the Group Chat is activated from the same Controlling Function as before (i.e. the session identity matches the one provided to the client), the Messaging Server Participating Function shall accept the session from the Controlling Function and connect it as a B2BUA to the one already established with the client. Furthermore the Messaging Server Participating Function shall on behalf of the user subscribe to the conference state provided by the Controlling Function and connect this subscription as a B2BUA with the subscription from the client that it terminated.

3.4.4.3.8 User explicitly leaving the Group Chat

When the user explicitly leaves the Group Chat, the Messaging Server Participating Function shall, as well as the forwarding of this action to the Controlling Function (as described in sections 3.4.4.1.2, 3.4.4.3.3 and 3.4.4.3.4 depending on the situation), also discard any remaining messages and notifications related to this Group Chat that were stored for delivery to that user.

3.4.4.4 Technical Realization of Full Store and Forward functionality for a Group Chat Participant

A Messaging Server may provide Full Store and Forward functionality for users participating in Group Chat. This feature applies no matter whether this Service Provider or another Service Provider is hosting the Group Chat.

3.4.4.4.1 Initiating a chat

Section 3.4.4.1.1 applies. When a user who is offline is invited to a group chat or a user does not respond to a SIP INVITE request for a group chat (i.e. the request times out), the

participating function will accept the group chat on behalf of the user and subscribe to the conference state information for that chat session. If it can handle File Transfer via HTTP, the participating function shall include the File Transfer via HTTP IARI in the Contact Header of the SIP 200 OK response sent when accepting the session and if no wildcard was used, also the *application/vnd.gsma.rcs-ft-http+xml* shall be included in the *a=accept-wrapped-types* SDP attribute that is included in that SIP 200 OK response. If it can handle Geolocation PUSH content, the participating function shall include the Geolocation PUSH IARI in the Contact Header of the SIP 200 OK response sent when accepting the session and if no wildcard was used, also the *application/vnd.gsma.rcspushlocation+xml* shall be included in the *a=accept-wrapped-types* SDP attribute that is included in that SIP 200 OK response.

Given that unless the user declines the Chat a Messaging Server Participating Function will accept the session, the Messaging Server Participating Function should anticipate to this and immediately accept the session from the controlling function without waiting for client's final response first. This behaviour will ensure that all messages sent in the Group Chat will ultimately reach the user also in exceptional circumstances where the Chat is closed by the Controlling Function before a final response was received from the user. When accepting the session before there is a final response from the user, the Messaging Server Participating Function shall in case a SIP 603 Decline response is received from the user terminate the session by sending a SIP BYE request to the Controlling Function carrying a Reason Header Field with the protocol set to *SIP* and the protocol cause code set to *200* (e.g. SIP;cause=200;text="Call completed").

3.4.4.4.2 Adding Participants to a Group Chat

The text in section 3.4.4.1.2 applies. If the Group Chat is a Closed Group Chat, the Messaging Server will return an error response to the SIP INVITE request.

Messages exchanged in the Group Chat before the new participant is invited are not subject to the Full Store and Forward functionality for that participant.

3.4.4.4.3 Closing Group Chat

The text in sections 3.4.4.1.3 and 3.4.4.3.8 applies.

3.4.4.4.4 Chat messages size limitations

The text in section 3.4.4.1.4 applies.

3.4.4.4.5 Delivery and Display notifications within Group Chat

The text in section 3.4.4.1.5 applies.

3.4.4.4.6 Interworking to SMS/MMS

The text in section 3.4.4.1.6 applies.

3.4.4.4.7 Restarting a Group Chat

The text in section 3.4.4.1.7 applies.

3.4.4.4.8 Storing messages for a participant who loses connectivity

The text in section 3.4.4.3.1 applies.

3.4.4.4.9 Rejoining a Group Chat after temporary disconnection, Group Chat is still ongoing

The text in section 3.4.4.3.2 applies.

3.4.4.4.10 Rejoining a Group Chat after temporary disconnection, Group Chat is over

The text in section 3.4.4.3.3 applies.

3.4.4.4.11 Storing messages for a participant who has not yet accepted the invitation

The text in section 3.4.4.3.1 applies with the difference that the Messaging Server Participating Function has been storing messages ever since the SIP INVITE request to the participant timed out and it became an endpoint on behalf of that participant by sending a 200 OK.

3.4.4.4.12 Joining and delivering messages for a participant who joins late, Group Chat is still ongoing

The text in section 3.4.4.3.5 applies with the difference that the Messaging Server Participating Function has been storing messages even if the participant had not joined before.

3.4.4.4.13 Joining and delivering messages for a participant who joins late, Group Chat is over

The text in section 3.4.4.3.4 applies with the difference that the Messaging Server Participating Function has been storing messages on behalf of the participant even if the participant had not joined before.

3.4.4.4.14 Inactive session from Controlling Function activated while forwarding messages

The text in section 3.4.4.3.7 applies.

3.4.5 NNI and IOT considerations

3.4.5.1 SIMPLE IM Group Chat – CPM Group Chat interworking

Interworking a participant with SIMPLE IM Group Chat towards a CPM Group Chat server, and a participant with CPM Group Chat towards a SIMPLE IM Group Chat server is done as per the [RCS5-CPM-CONVFUNC-ENDORS], via mapping of the appropriate Chat session feature tags when it is determined that the remote network requires such interworking.

3.4.5.2 Messaging Server handling of delivery notifications when not supported by device or terminating network

For devices or networks (e.g. older RCS versions) that the controlling or participating function of an RCS Messaging Server recognizes do not support generation of delivery notifications within a Group Chat, the Messaging Server shall generate them on behalf of the devices if such notifications are requested. As it is related to the deployment environment, the method by which the Messaging Server determines whether a device can generate delivery notifications is considered outside of the scope of this specification. The following may be reliable indications though:

- The device indicates explicitly support for the *message/imdn+xml* Mime type in the *a=accept-wrapped-types* SDP attribute.
- The device indicates support for File Transfer via HTTP in the Group Chat (see section 3.5.4.8)

A device that does not generate delivery or display notifications should not receive the CPIM IMDN header in an MSRP SEND that is used to request notifications. When sending a message to a device or network that is assumed not to support disposition notifications in Group Chat, the Messaging Server shall remove this header.

3.4.5.3 Interworking between Chat participants not allowing multimedia content and Chat participants allowing multimedia content

Whether multimedia is allowed in a Group Chat session depends on whether it is supported by the initiator of the session and therefore on the policy of the operator hosting the controlling function in the Group Chat. If it is not allowed, the *a=accept-wrapped-types*

attribute in the SDP in the SIP INVITE request shall only include *text/plain*, *message/imdn+xml* and *application/im-iscomposing+xml*. If File Transfer using HTTP is supported (see section 3.5.4.8) then the a=accept-wrapped-types attribute shall also include *application/vnd.gsma.rcs-ft-http+xml*. If Geolocation PUSH is supported (see section 3.10.4.1.3), then the *a=accept-wrapped-types* attribute shall also include *application/vnd.gsma.rcspushlocation+xml*. If multimedia is supported more types can be listed or a wildcard can be provided. An invited client can therefore learn from the received SIP INVITE request whether multimedia is supported in the Group Chat session.

If multimedia is supported in the Group Chat session, a client that is invited can still be configured by an operator not to support multimedia in chat sessions. In that case it shall only include *text/plain*, *message/imdn+xml* and *application/im-iscomposing+xml* in the SDP in the SIP 200 OK response when accepting the session indicating to the controlling function that such content should not be distributed to that participant. If such content is then sent in the Group Chat, the controlling function could alert the sender and the participants not supporting the content through a system message as defined in [RCS5-SIMPLEIM-ENDORS] of this situation. The controlling function shall not send the content to those recipients not supporting it.

This occurs in NNI situations when one user is served by a network supporting multimedia content in Chat sessions and another user is served by a network supporting text only in Chat sessions. A Messaging Server where only text is allowed in Chat sessions shall ensure that the *a=accept-wrapped-types* attribute in the SDP used to negotiate the content in the MSRP only contains the *text/plain, message/imdn+xml and application/im-iscomposing+xml* content-types and if File Transfer using HTTP or Geolocation PUSH is allowed (see sections 3.5.4.8 and 3.10.4.1.3) *application/vnd.gsma.rcs-ft-http+xml* and *application/vnd.gsma.rcspushlocation+xml* respectively.

3.4.6 Implementation guidelines and examples

Please note that the following sections propose an experience which is aimed to be employed as a reference for OEM implementations.

3.4.6.1 Protocol flow diagrams

NOTE: To simplify the use cases including the store and forward function the figures and text do not differentiate between different Service Providers and their message servers. These servers are combined to one generic message server function to focus on the client/server communication aspects.

3.4.6.1.1 Start a Group Chat from the Chat application

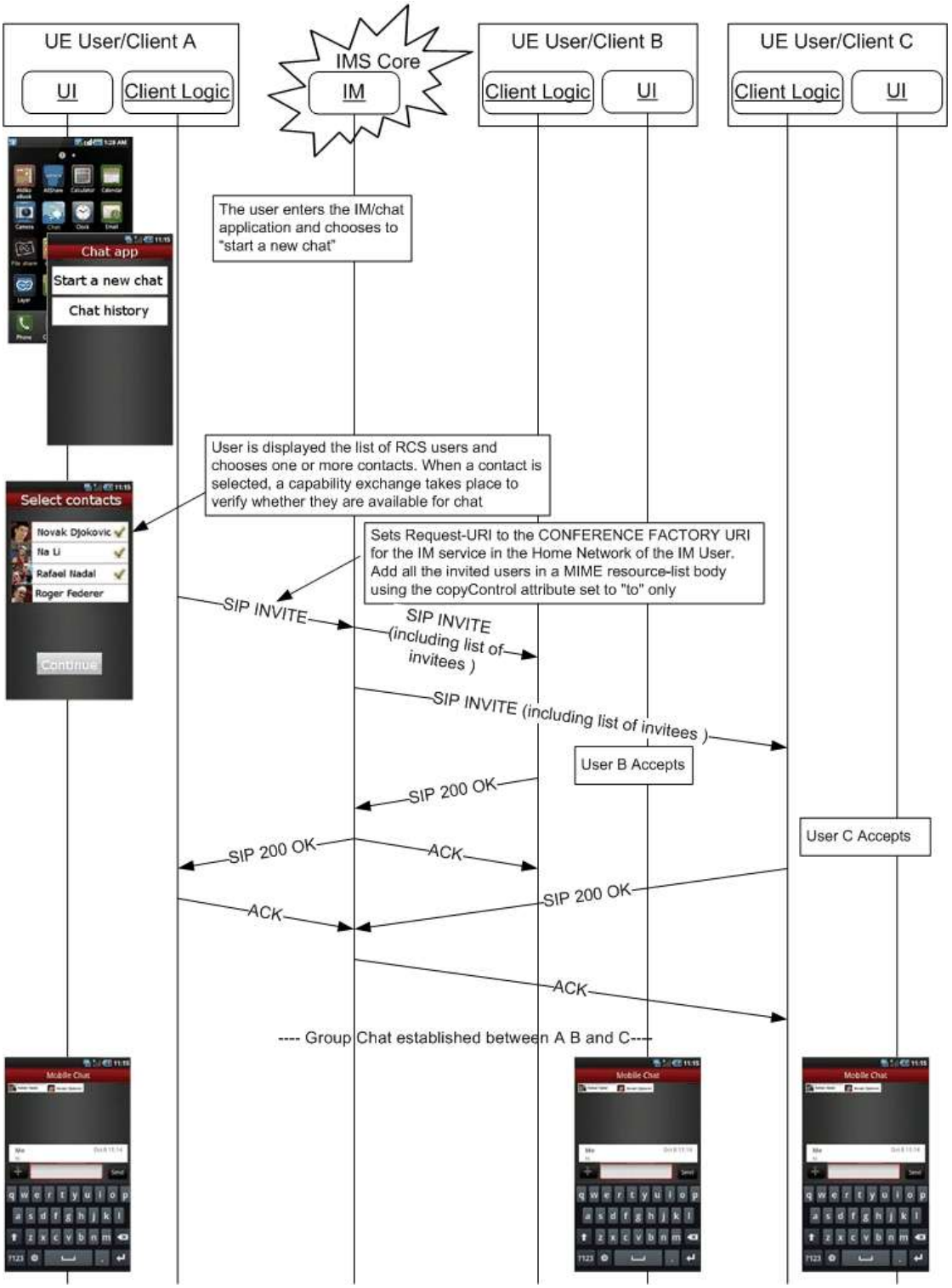


Figure 49: Start a Group Chat from the Chat application

NOTE: The above flow mentions that OPTIONS is used for service capability exchange, which is the case when the DEFAULT_DISCOVERY_MECHANISM is set to 'OPTIONS'. When it is set to 'Presence', then Presence is used.

The UX associated with an RCS Group Chat should provide the following functionality:

- Displaying the list of participants of the current Group Chat and providing of notifications when a new participant is joining and when a participant is leaving the current Group Chat
- When starting a Group Chat session, the invitation shown to the invited users should list the participants invited to the Group Chat before accepting the invitation (e.g. "You're invited to a group chat with A, B & C" instead of "A is inviting you to a group chat")

3.4.6.1.2 Start a Group Chat from the Chat composition window

In this case, Users A and B are in a chat, and User A decides to add a third user (User C) to the chat session. The relevant UX and flow sequence is presented below:

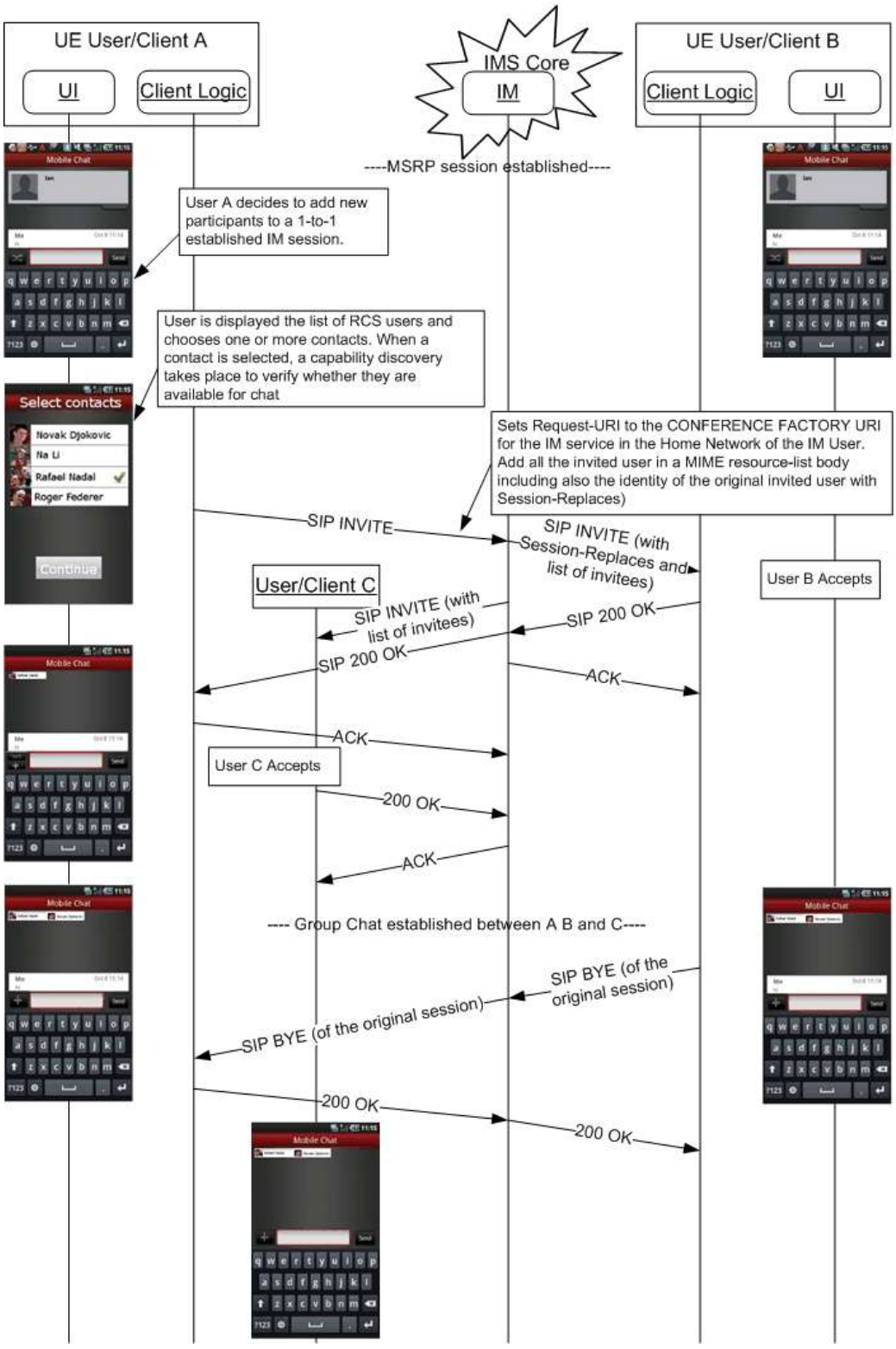


Figure 50: Group Chat session initiation

NOTE: The above flow mentions that OPTIONS is used for service capability exchange, which is the case when the DEFAULT_DISCOVERY_MECHANISM is set to ‘OPTIONS’. When it is set to ‘Presence’, then Presence is used.

3.4.6.1.3 Get participants of Group Chat

The following flow is complementary to the previous use case as it presents in detail how to get information on the chat participants. Please note that these exchanges were omitted in the previous diagram:

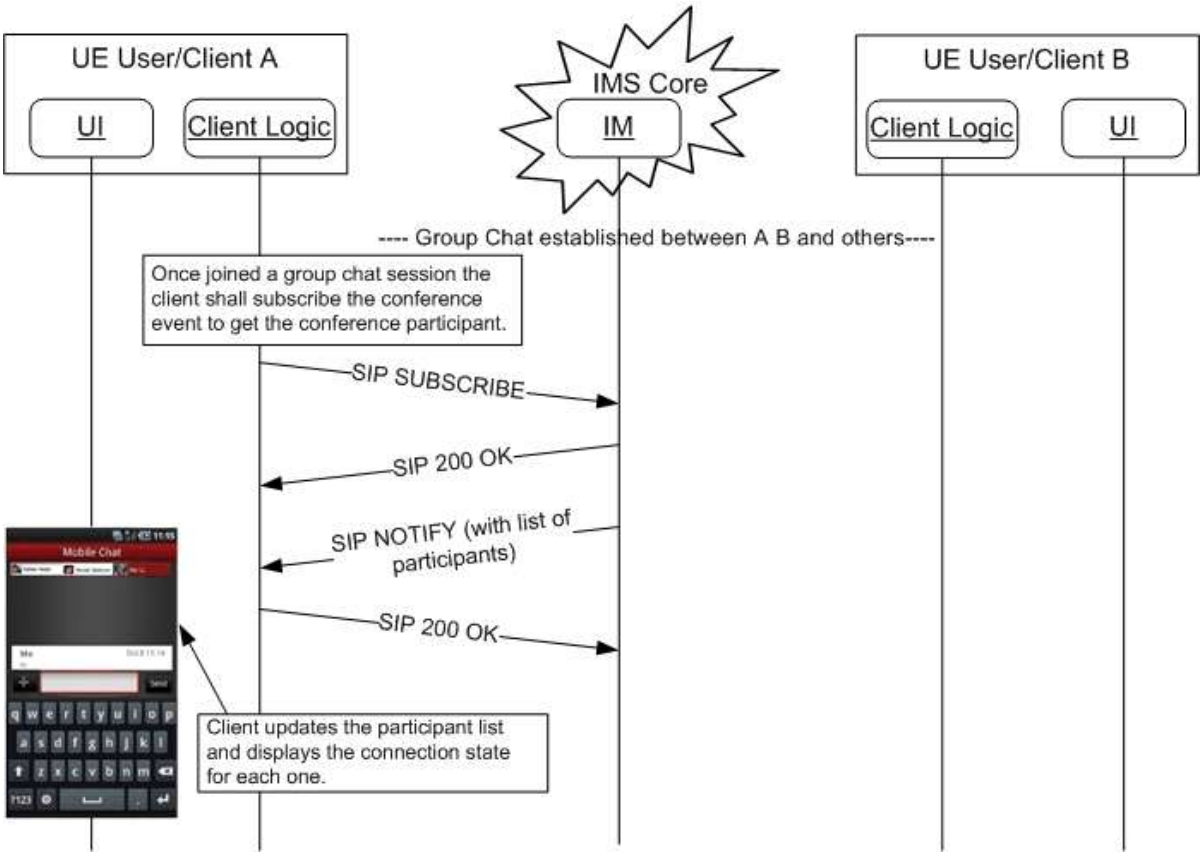


Figure 51: Group Chat session initiation (II): Get participants

3.4.6.1.4 Add a participant to an already established Group Chat

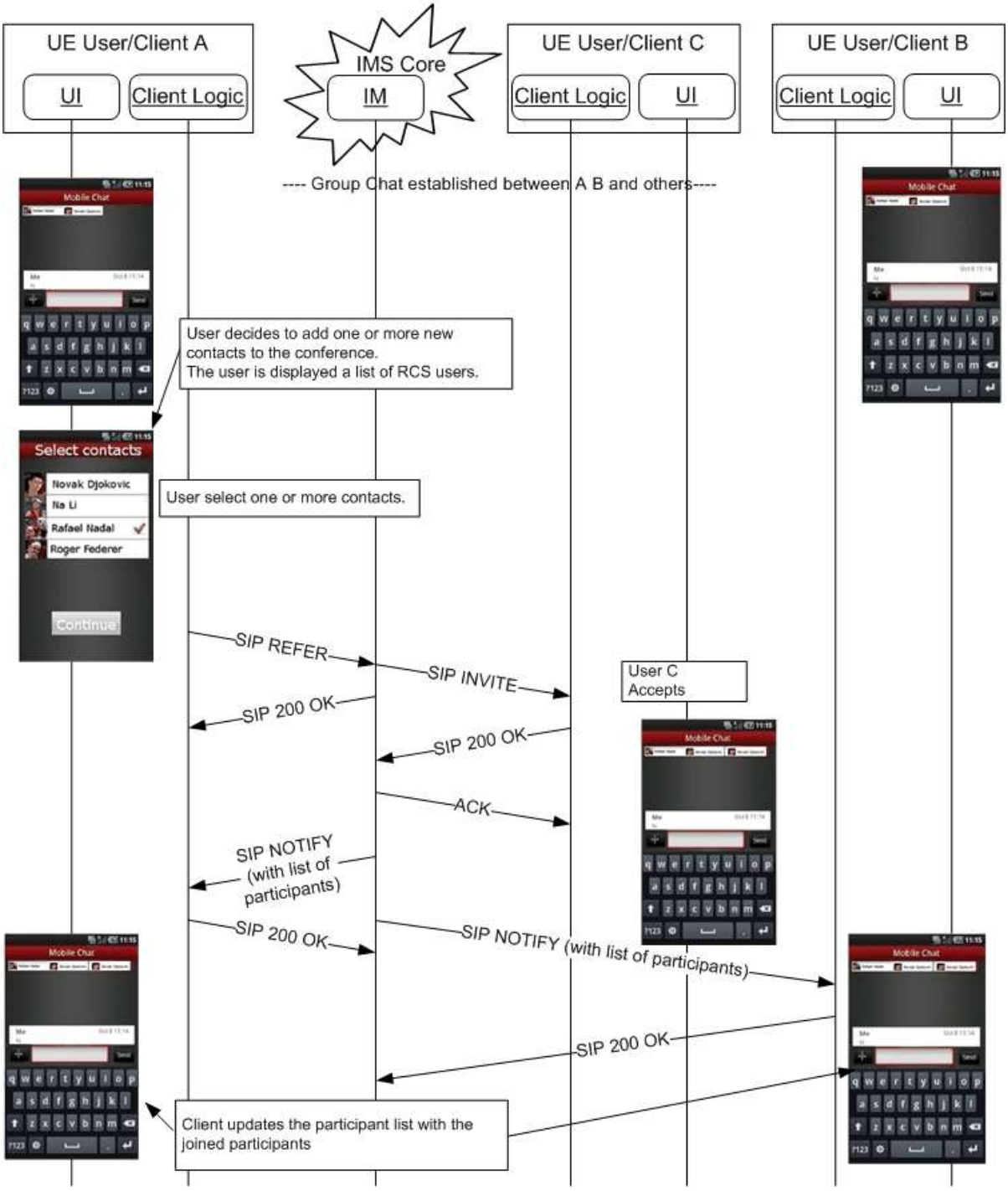


Figure 52: Adding new users to a Group Chat

3.4.6.1.5 Sending a Chat message from the Group Chat window

NOTE: the flow does not show Client B and Client C generating a delivery notification for the received chat message; however it is expected that they generate one if it was requested.

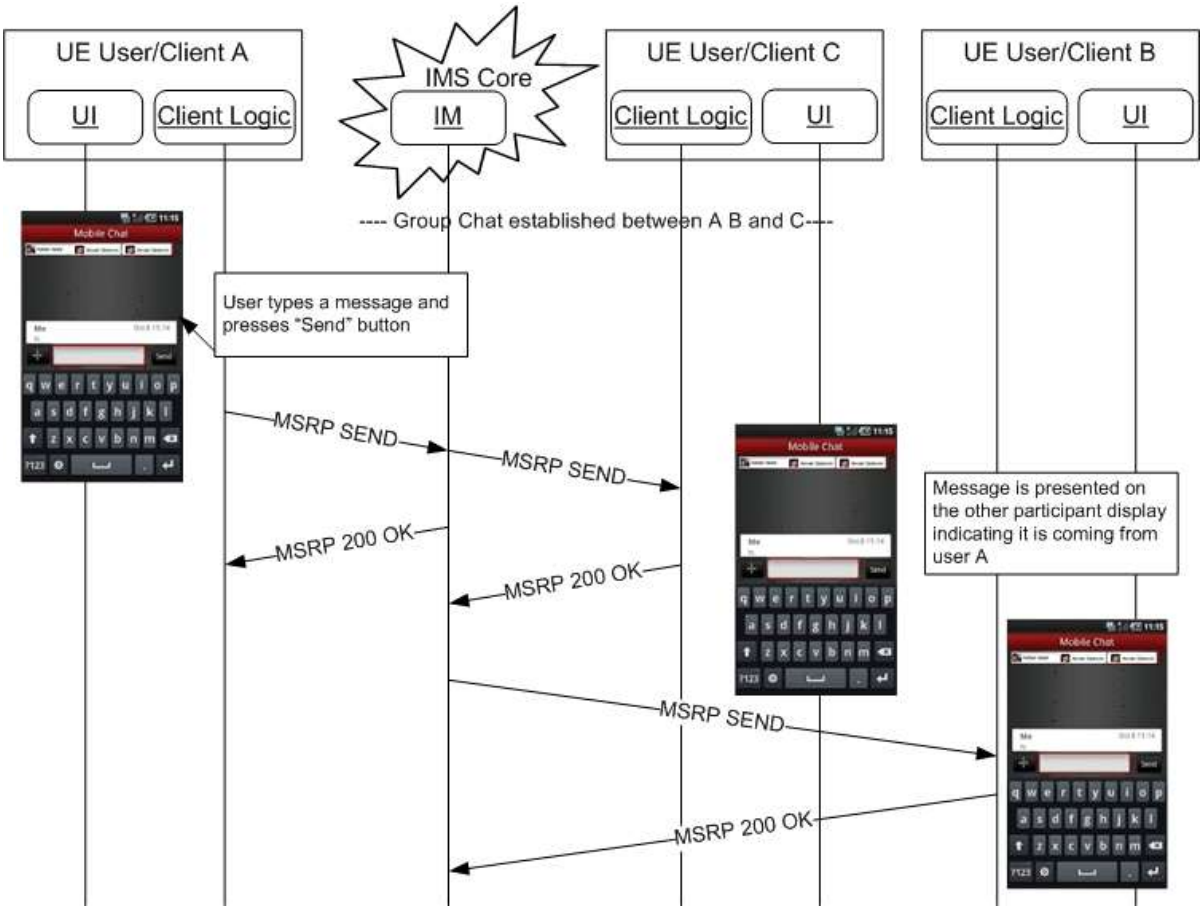


Figure 53: Chat message sequence for a Group Chat

NOTE1: As described in section 3.5.4.8.1, if the message that is exchanged is a File Transfer via HTTP body, the conference focus shall not forward the body to participants that haven't indicated support for File Transfer via HTTP.

NOTE2: As described in section 3.10.4.1.3.2, if the message that is exchanged is a Geolocation PUSH body, the conference focus shall not forward the body to participants that haven't indicated support for Geolocation PUSH.

3.4.6.1.6 User in a Group Chat goes offline

In the following flow, Users A and B are in a chat among others (Group Chat); Basic or Full Store and forward is not provided for User B; suddenly User B goes offline (due to the loss of the connection to the network for example):

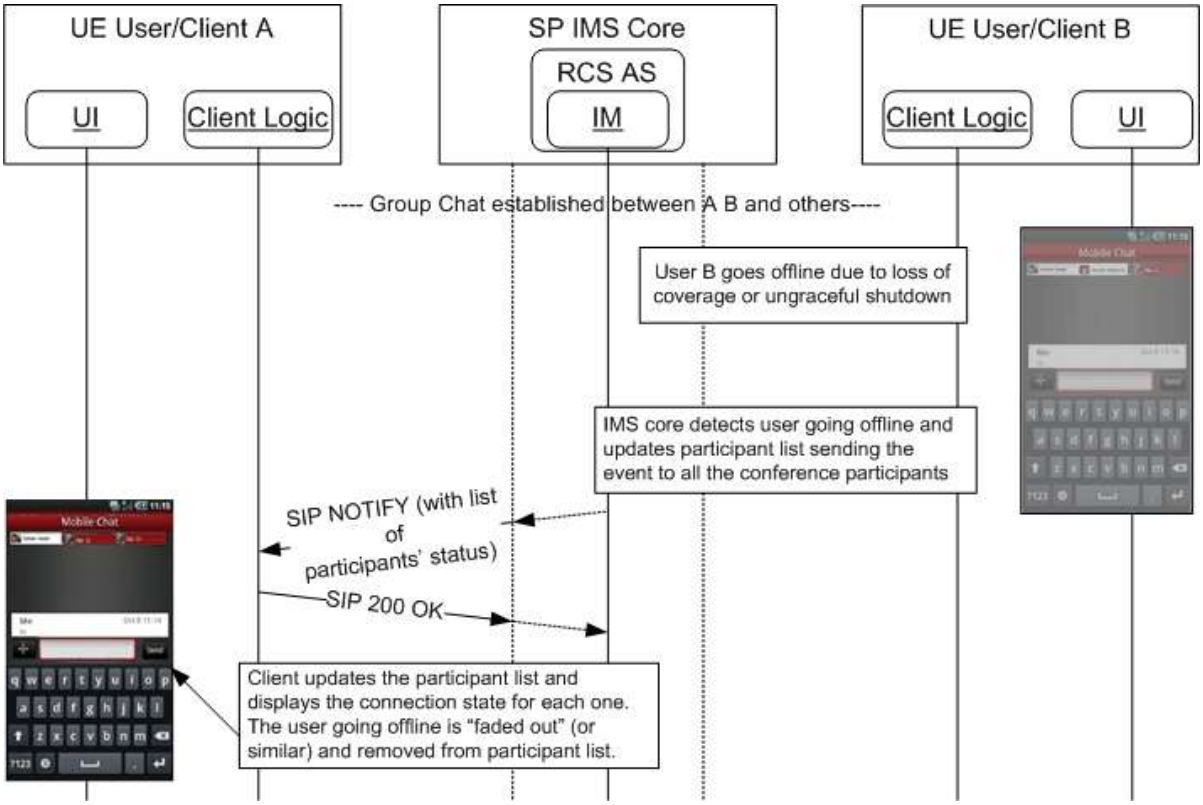


Figure 54: Forced chat termination in a Group Chat: User goes offline

In this case, User B's device should store the Group Chat's IM Session identity for some time and when it regains connectivity it should automatically attempt to rejoin using the procedures described in [RCS5-SIMPLEIM-ENDORS] or [RCS5-CPM-CONVFUNC-ENDORS]. When that attempt results in a SIP 404 Not Found response, the chat should be considered to be no longer active and no further attempts to rejoin shall be performed, but a new group chat shall be established as per section 3.4.4.1.1, using the last participant list it had already stored to build the URI-list in the SIP INVITE request.

3.4.6.1.7 Leaving a Group Chat

This case is equivalent to the previous one. In this case however, User B leaves the chat intentionally:

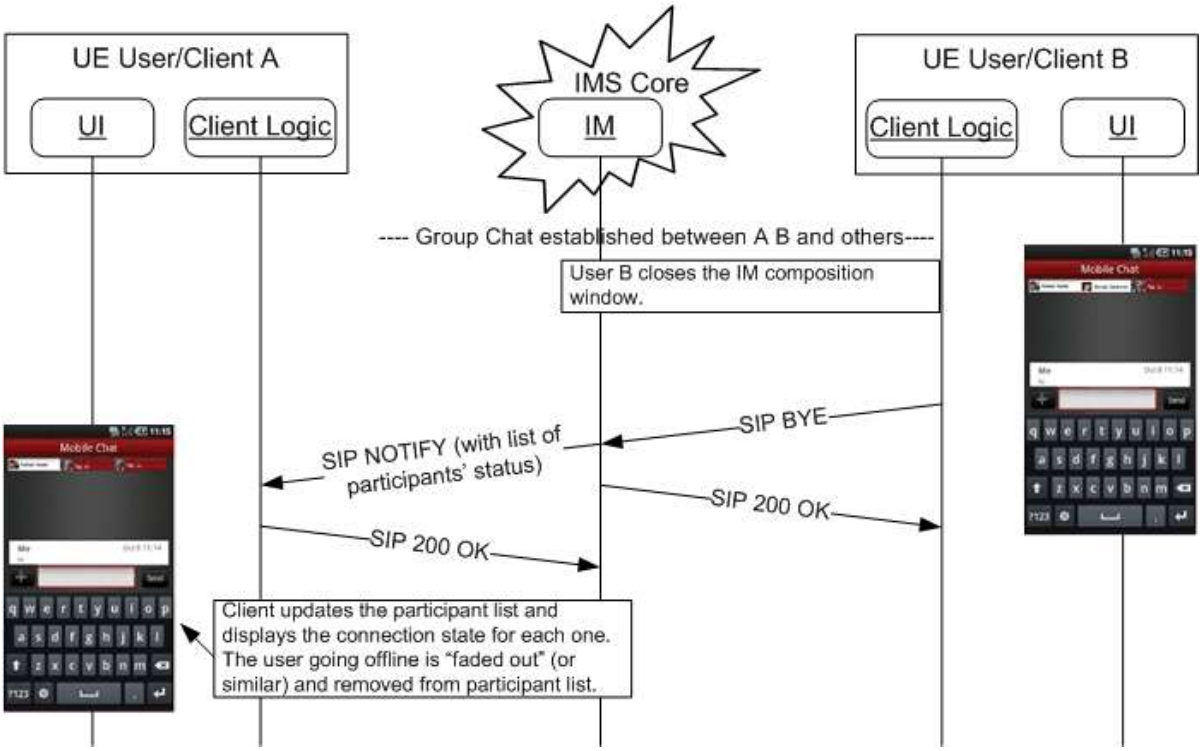


Figure 55: Leaving a Group Chat

3.4.6.1.8 Setting up a Closed Group Chat

In the following flow, User A initiates a Closed Group Chat with Users B and C, but does not invite User D;

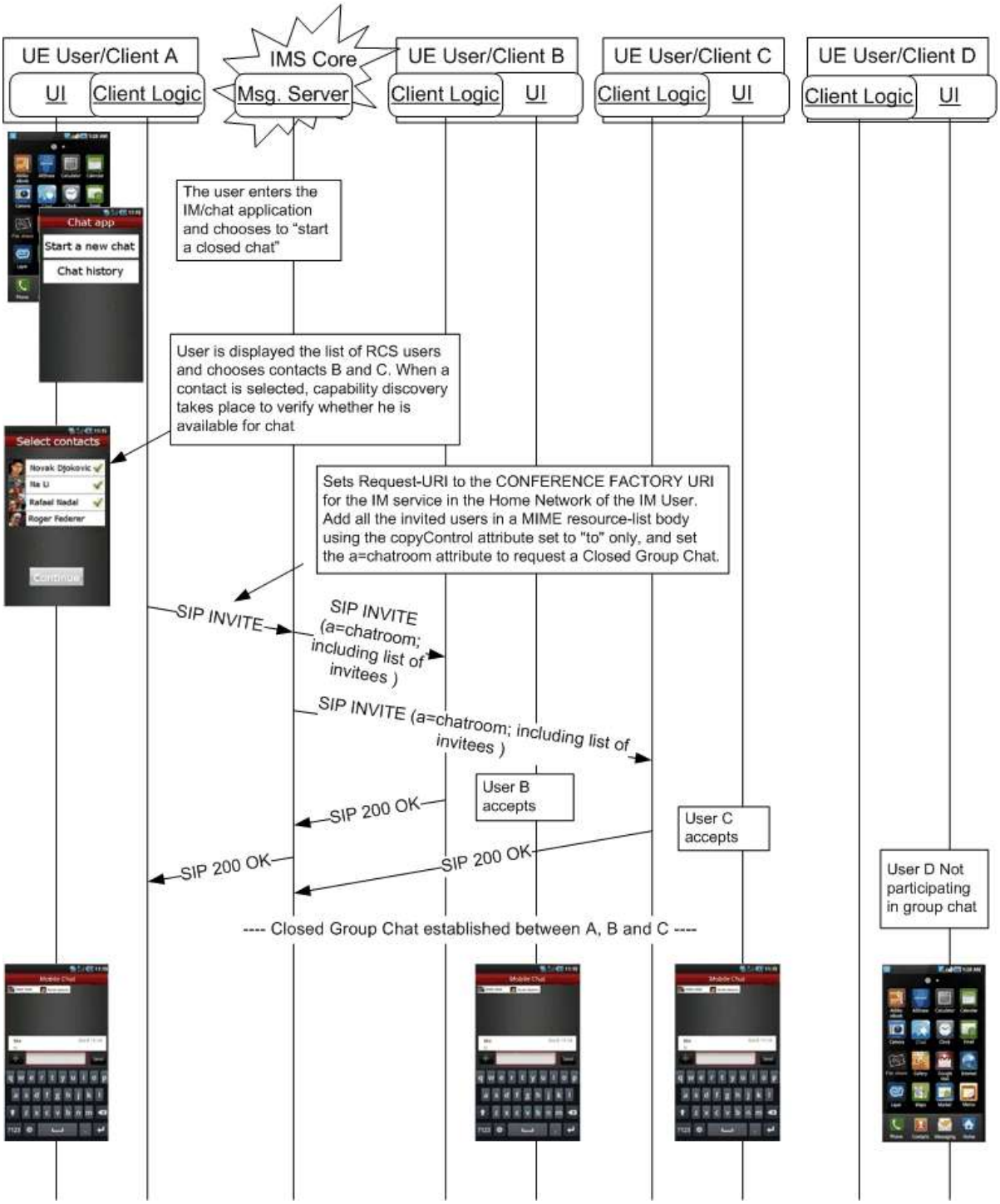


Figure 56: Setting up a Closed Group Chat

3.4.6.1.9 Add new participant for a Closed Group Chat

In the same Closed Group Chat as in Figure 56, User B decides to invite User D, selecting from the list of contacts. Since this is a Closed Group Chat no additional participants can be added and the Messaging Server will return an error:

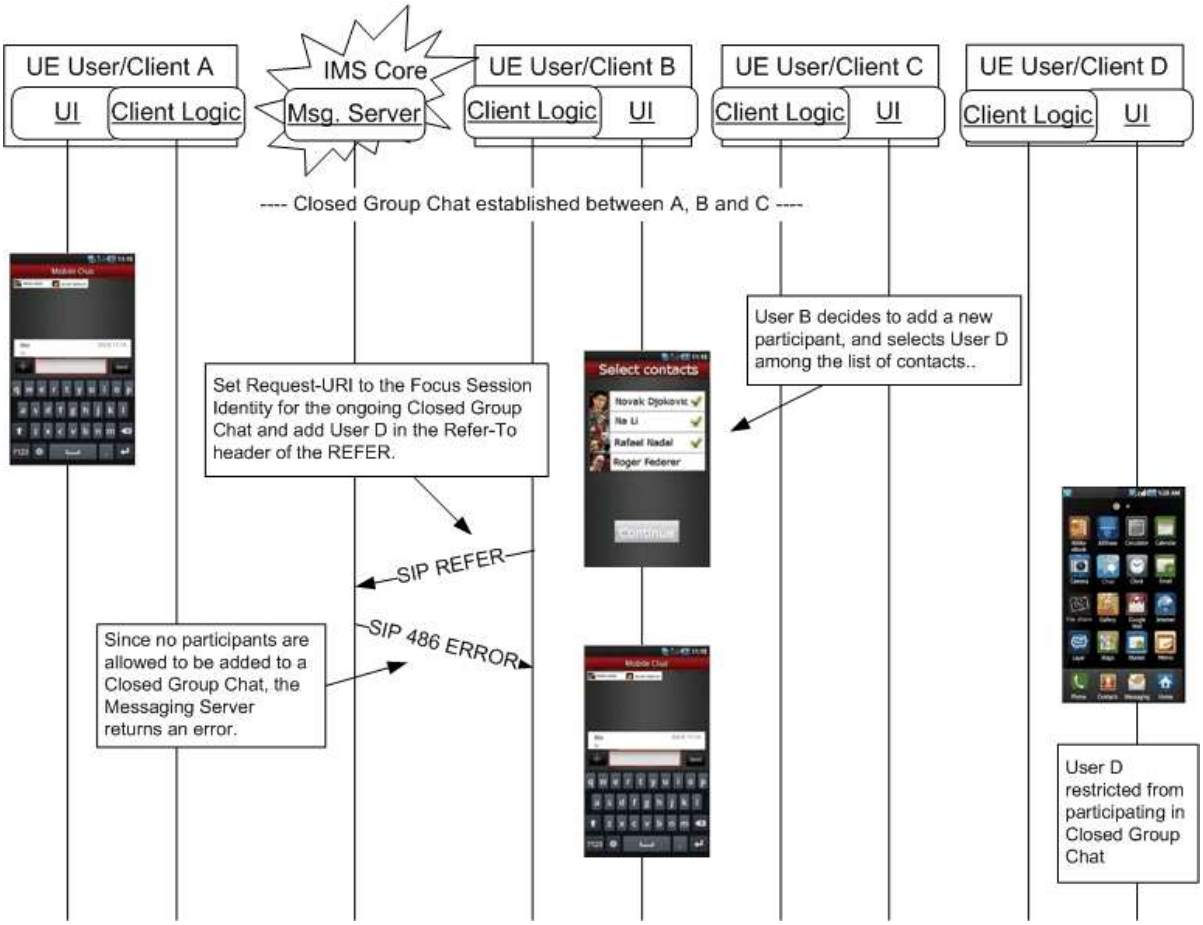


Figure 57: Add new participant for a Closed Group Chat

3.4.6.1.10 Setting up a Group Chat with Store and Forward support

In the following flow, User A initiates a group chat with Users B, C and D. User B subscribes to Full Store and Forward with Auto-accept capabilities, but decides to join the group chat later. User C manually accepts the group invitation. User D subscribes only to Basic Store and Forward and joins manually.

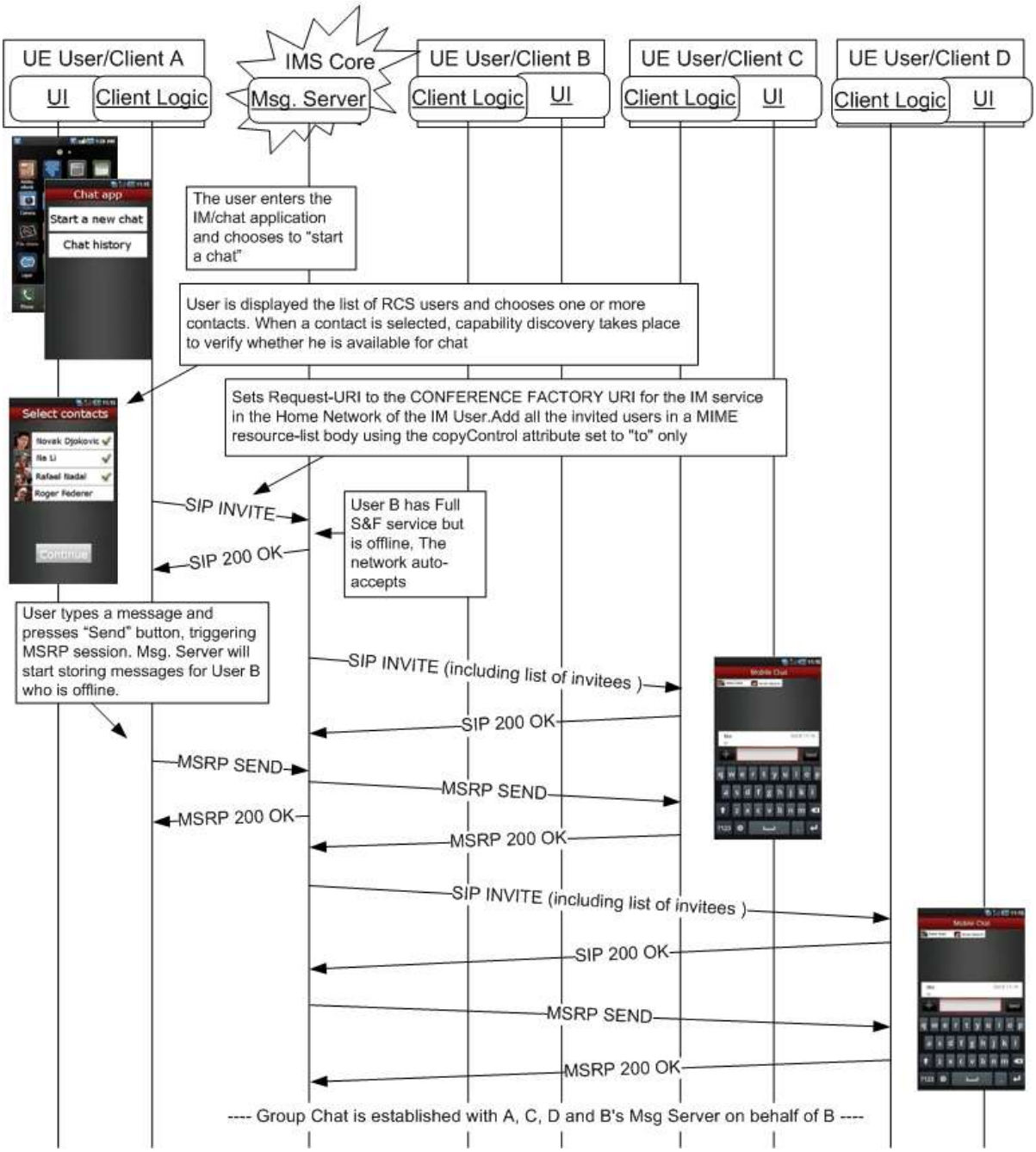


Figure 58: Setting up a Group Chat with Store and Forward support

Figure 59 shows the flow when User B joins the chat:

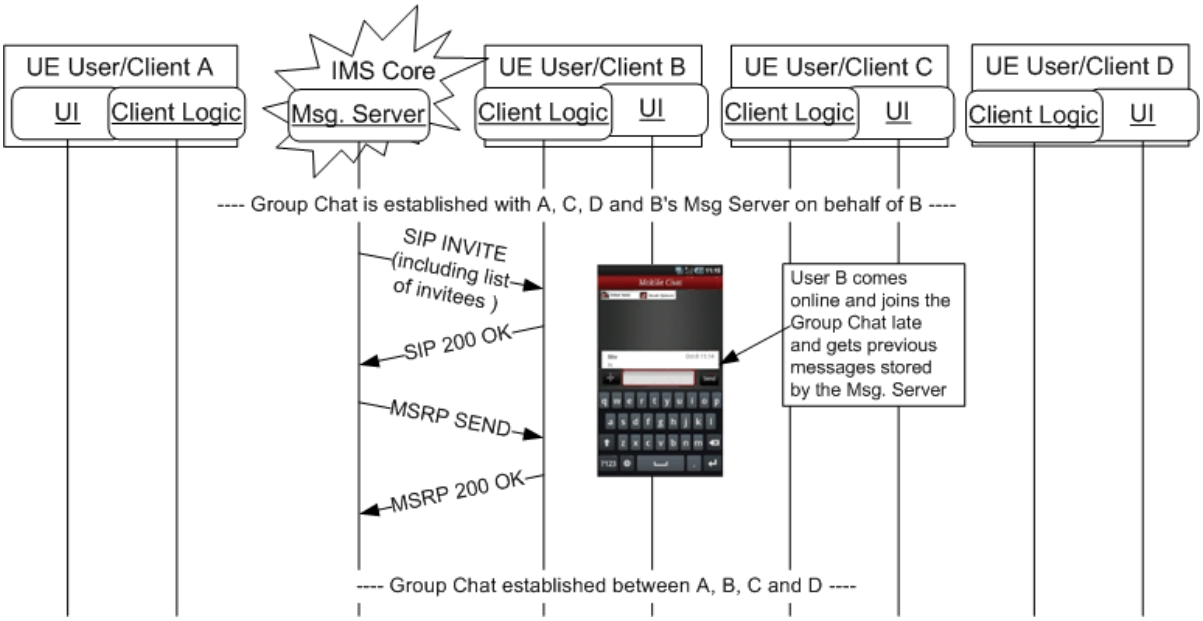


Figure 59: User with full store and Forward joins late

3.4.6.1.11 User in a Group Chat goes offline when Messaging Server supports Store and Forward

In the following flow, Users B and D are in a chat among others (Group Chat); suddenly Users B and D go offline (due to the loss of the connection to the network for example). Messaging Servers supporting Users B and D store subsequent messages received for Users B and D:

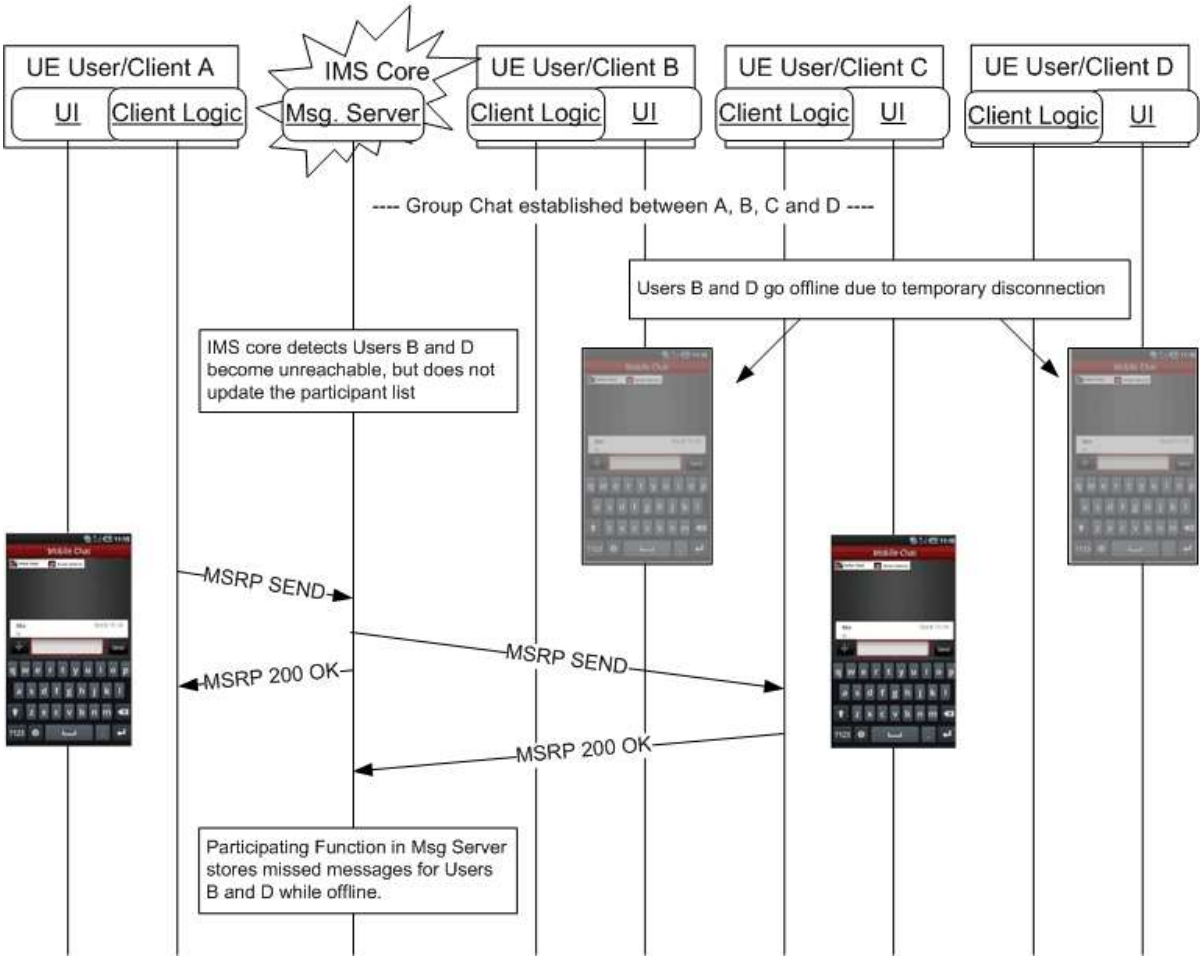


Figure 60: Users go offline when Messaging Server supports Store and Forward

3.4.6.1.12 Rejoining a Group Chat after temporary disconnection, Group Chat is still ongoing

In the same Group Chat as in Figure 60, Users B and D are back online and the Messaging Server will deliver the messages it has stored for both users, and the Group Chat is resumed with all original participants:

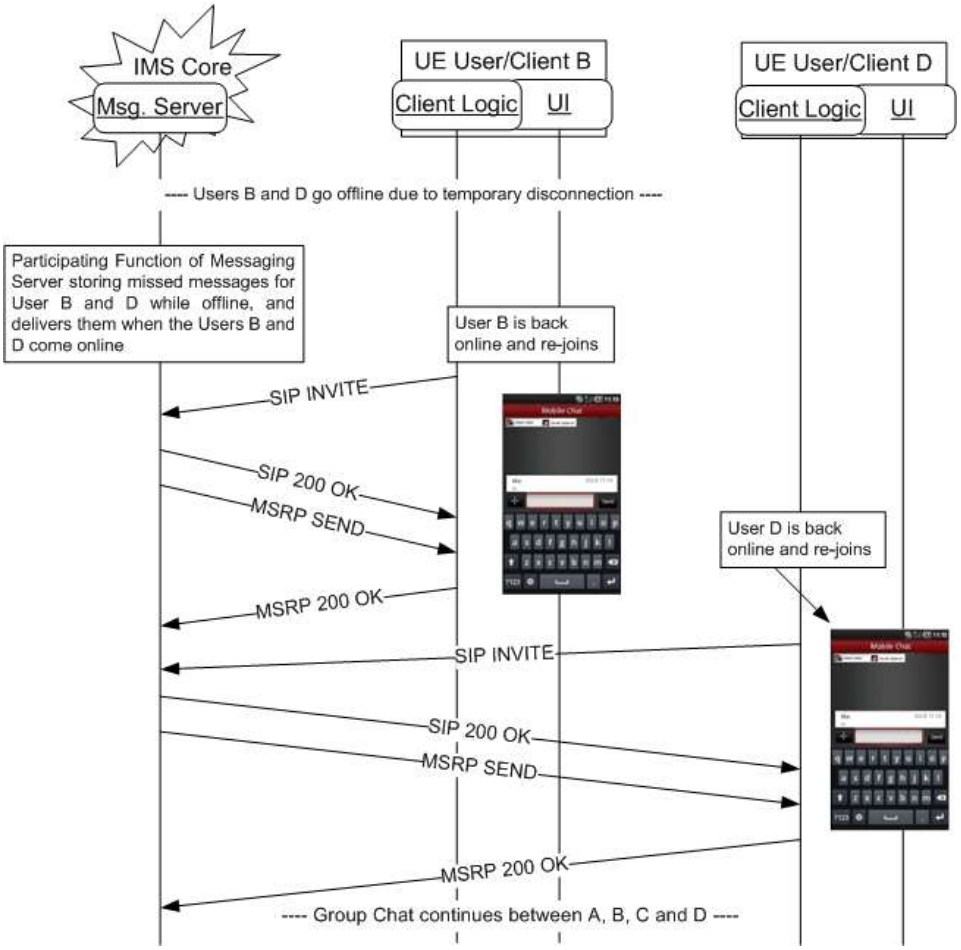


Figure 61: Rejoining a Group Chat after temporary disconnection, Group Chat is still ongoing

3.5 File Transfer

3.5.1 Feature description

File Transfer is the ability for users to exchange different types of content (files), during an ongoing session or without having an ongoing session.

On the sender's side, before sending the request to the intended recipient, the file to be transferred and the recipient have to be selected (refer to use cases in section 3.5.6). If the recipient is not registered, and if the recipient has the File Transfer store and forward service capability, it may still be possible to send the file transfer request (refer to use cases in sections 3.5.4.7 and 3.5.4.7.2.1).

For pictures or video clips it is a significant added value if the recipient receives a preview of the proposed file before accepting or declining the transfer. Therefore, whenever possible, the sender of the file should include a thumbnail of the file in the File Transfer invitation. A client receiving a File Transfer request with a thumbnail should display the thumbnail in the pop-up presenting the File Transfer invitation.

The request for File Transfer is sent to all of the recipients' devices. When no automatic acceptance is done, this will trigger a pop-up indicating to the user that a contact wishes to send them the depicted file. The recipient is able to select the device to which the file is transferred by accepting or refusing the File Transfer invitation on that device.

In this pop-up shown prior to the actual transfer of a file, the intended recipient is given the opportunity to learn about the proposed File Transfer (size, name, preview and type of file in addition to the identity of the sender) and then to accept or decline the File Transfer based on this information.

If a File Transfer is interrupted for any reason, the receiver can request resumption of the File Transfer without having to re-start from the beginning.

Users are allowed to qualify undesired incoming File Transfer requests as spam. To this end, clients may support a locally stored black list to handle incoming File Transfer requests. This is the same black list as it is used for incoming chat requests. If an invitation to receive a file is received from a blacklisted user, the client should reject the File Transfer request, and from the UX, not notify the user. Instead it may log the event in the spam folder (e.g. "User A tried to send a file on TIME/DATE").

The File Transfer feature has the following limitations:

- Sharing files with a group of users is only considered within a Group Chat Session. Outside of a Group Chat Session, a device UI may initiate multiple 1-to-1 File Transfer sessions to transfer a file to multiple users**.**
- Only one file can be sent per file transfer session.

3.5.1.1 Handling of specific content

For some of the content exchanged during a file transfer specific handling is provided. This is described in the following subsections

3.5.1.1.1 Card Push

Sharing contact information brings different opportunities to RCS, all of them increasing end user contact possibilities e.g. allowing RCS Users to connect with other RCS or non-RCS Users.

Currently manufacturers are saving the contact info in their address books without following a fully open standard and, as a consequence, sharing this information effectively with other device manufacturers becomes a challenge.

Also, the concept of 'personal' and 'business' card, representing the user's own contact information, which may be stored in the address book, is not used simply because this is not an explicit option of the address book menus of existing devices.

This specification aims to:

- Move towards a standard format compliant with all kinds of devices for keeping contact information.
- Create and manage personal and business cards and share them with selected contacts and giving this option visibility in the address book menus.
- Exchange contact information in a secure way.

RCS brings File Transfer as a new service, which becomes a very good bearer for exchanging of contact cards among users. Those contact cards can be sent to another user, like any other file format, using File Transfer.

3.5.1.1.2 Audio Message

The Audio Messaging feature is described in section 3.11.

3.5.2 Interaction with other RCS features

A 1-to-1 File Transfer is not linked to other services (for example CS-voice call or ongoing chat session) and can be used either during or outside of other communication sessions. The procedure for any file transfer within an ongoing 1-to-1 chat session is implemented as a separate session in parallel with the ongoing 1-to-1 chat and therefore is the same as the procedure for initiating a separate session for File Transfer.

Different types of content (files) can be exchanged during an ongoing session or without having an ongoing session, i.e., during or outside a call or 1-to-1 chat session.

When transferring a file while not in an existing session (that is when not in a call or chat session with the contact with whom the file is to be shared) and after the transfer has started (that is the receiving user accepted the incoming file) the file transfer is presented to the recipient in a chat context. This establishes a communication context for the transfer since the recipient may want to know why the sender is sharing the file. At the time the file is presented, the chat session is not started. The chat session will only start if and when the receiver sends a chat message back to the sender of the file transfer.

Figure 62: Reference UX for file transfer on the receiver side

When a file transfer is started during a call with the receiver of the file transfer, the file transfer continues until it is completed or cancelled, i.e., the file transfer will not be terminated when the call ends.

A File Transfer is possible during a group chat. In this case the file is sent to all participants that are capable of receiving the file.

3.5.3 High Level Requirements

3-5-1 Files can be exchanged during a session (e.g. CS voice call or message conversation)

3-5-2 A File Transfer can be initiated by either end point while having an ongoing session (e.g. the caller or the callee)

3-5-3 End of file transfer shall not lead to termination of a simultaneous ongoing session

3-5-4 End of a voice call shall not lead to termination of ongoing file transfer

3-5-5 Files can be exchanged without having an earlier established session (e.g. directly from a multimedia gallery).

3-5-6 The receiver must be able to accept or reject offered files. The invitation procedure shall include an indication to the receiving party concerning file size and type.

3-5-7 The receiver shall have the possibility to save the transferred files.

3-5-8 It shall be possible to assign a service provider configurable maximum file size allowed for File Transfer.

3-5-9 The sending and receiving client shall be able to resume an interrupted file transfer. It is up to Service Provider policy whether only the receiving client or either client can initiate the resume request.

3-5-10 The sending client shall have the possibility to include a thumbnail preview of an offered file.

3-5-11 When sending a file to the recipients in a group chat, the file shall be sent to the network only once.

3-5-12 Store & forward: If the intended recipient is not available, or the recipient does not accept the file transfer invitation within a Service Provider define time limit, the file being offered for transfer shall be available for later retrieval, provided the recipient also has the store & forward service and it is enabled (determined by capabilities exchange).

3.5.4 Technical Realization

File Transfer is based on [RCS5-SIMPLEIM-ENDORS] and [RCS5-CPM-CONVFUNC-ENDORS], as well as on the extensions described in [RFC5547]. The technology choice is controlled by the configuration parameter CHAT MESSAGING TECHNOLOGY as described in section A.1.3.3.

SIP INVITE requests for file transfers will be forked to all the recipient's devices. If the recipient accepts the invitation on one device, the corresponding client shall respond with a 200 OK response. If the recipient rejects the session, the client shall respond with a 603 response. In both cases, the IMS network of his Service Provider will cancel the invitations to his other devices.

The SIP 603 Decline response shall be used for the automatic rejection of the incoming File Transfer invitation in case the initiator is included in the device's local blacklist that is described in section 3.5.1.

The current solution provides two complementary technical realizations to provide the file store and forward functionality:

- File fetching/re-delivery via SIP and MSRP: In this implementation, the receiver shall either fetch the stored file using the file transfer fetch procedure described in [RFC5547], or send a new invite to the MSRP server that stored it or wait for a new invitation for the file transfer to be sent by the server that has stored the file previously when the user was offline when the transfer was initiated.
  This is described in sections 3.5.4.2 and 3.5.4.7
- File fetching via HTTPS: In this technical realization, the receiver shall fetch the deferred file using an HTTPS request.
  This is described in section 3.5.4.8.

The two solutions are complementary; therefore, a service provider can choose the best combination to provide the file store and forward service to their customers maximizing the resources that are already deployed in their networks.

The recipient's client may depending on the setting of client configuration parameter FT AUT ACCEPT (See Table 86 in Section A.1.4) automatically accept the File Transfer invitation from users not included in the device's local blacklist provided the size indicated in the SDP is below the maximum size for the file transfer warn size (FT WARN SIZE) (See Table 86 in Section A.1.4). Please note in roaming scenarios auto-acceptance shall be disabled by default and if the FT AUT ACCEPT parameter is set, the RCS client shall provide a configuration setting to the user so it can be enabled.

If FT AUT ACCEPT (See Table 86 in Section A.1.4) is set to disable automatic acceptance (i.e. set to a value of 0), File Transfer shall be never auto-accepted.

NOTE: a Service Provider should take into account that enabling any auto-acceptance feature, like the one described above, will impact the multidevice behaviour as it may lead to race conditions.

The extensions described in [RFC5547] are used as follows:

- The SDP payload for File Transfer requests is populated according to [RFC5547], i.e. both sending and receiving clients need to support all elements of [RFC5547]. For populating the file-selector attribute, it is preferable to use the hash-selector, in addition to the other selectors possible. The reason being that the hash-selector uniquely identifies a file, and can also be used to verify the correct transfer of the entire file. The SDP payload shall contain the file size.
- An interrupted file transfer can be resumed by the recipient sending a new SIP INVITE to the originator asking for the missing part of the file. For this it uses the file-range attribute (to denote the missing part) including the file-selector (to denote the file). Note that absence of the file-range attribute denotes transfer of the entire file.
  For such a pull-style operation, the SDP attributes, including file-range and file-selector are populated as described in [RFC5547]. Especially note that from the viewpoint of [RFC5547] this is a new file transfer and hence it will carry a new file-transfer-id attribute.
  To support multiple devices on the originating side, the file recipient should address the originating RCS UA via device identifier (sip.instance or GRUU, see section 2.11.3) to be able to resume the file transfer at a later stage. If the device identifier of the file sender is included in the initial SIP INVITE received by the file recipient, it has to be included by the file recipient in the new SIP INVITE sent to the originator. If the device identifier is not included in the SIP INVITE received by the file recipient it cannot be included in the new SIP INVITE, and the SIP INVITE will be forked to all the registered devices of the originator. In that case, any device which has stored the requested file will answer the SIP INVITE with 200 OK if accepted by the user or the RCS client. For security reasons, an auto-acceptance of resumption requests shall only be offered when a clear correlation between the initial file transfer and the related resumption request can be ensured by the client implementation. In case of manual acceptance, the RCS client application may notify the user that this is a file pull for sake of a resumption request (rather than an ordinary file transfer).
- Generic file pull scenarios (as described in [RFC5547]), i.e., scenarios that do not pursue on a preceding file transfer as described above, are not supported in this specification.
- In scenarios where the file sender notices that an initiated file transfer could not complete successfully, such an interrupted file transfer can also be resumed by the file sending client.
  - The procedure for resumption by the file sending client corresponds to the resumption by the file receiving client described above except for the following differences:
    - The file sending client will send a new SIP INVITE request with a file selector and a proposed file range in the SDP based on information the file sending client has upon detection of the failure condition.
    - The file receiving client will use the file selector and the file range attribute to determine it is a resume request (for this the receiving client may keep information of interrupted file transfers). The file receiving client should include the exact file range required in the SDP returned in the 200 OK on the SIP INVITE request initiating the resumption.

    - Upon reception of the SDP in the 200 OK, the file sending client shall always use the file range specified by the file receiving client for the resume operation.
  - If the file receiving client does not support resumption, the SIP INVITE for the resume will be rejected. The file sending client that initiates the resumption should not continue the resumption. Alternatively, it might then re-send the entire file.
  - For the case where the file recipient user has multiple devices, the file recipient client needs to address the correct file sending client by using the device identifier (sip.instance or GRUU) it received in the original file transfer invitation response if any as described in section 2.11.3.
  - If both clients initiate resume, the file recipient's request should be given preference since the file recipient has accurate information about the missing parts of the file. This means that in that case the file recipient client will decline the SIP INVITE request issued by the file sending client.
- If the contact has indicated the capability for receiving a thumbnail and FT THUMB (see section A.1.4) is set to enabled, a preview of an offered file can be added to the SDP description of the SIP INVITE request by using the file-icon attribute of [RFC5547]. The size of this thumbnail shall be smaller than 10 kB. Other SDP attributes will be populated as described in [RFC5547].
  - The procedure describing how to create the thumbnail itself, in its raw binary form, is out of scope of this specification. For a picture, the raw binary result shall be a thumbnail of the picture itself. For a video clip, the raw binary result shall be a thumbnail either of the first I-Frame at 20% of the total length of the video clip or of another relevant frame.
  - The size of a thumbnail should be restricted to the minimum number of octets that provide significance.

In the following sections, the relevant message flows and reference UX are shown. These are based upon the following assumptions:

- For simplicity, the internal mobile network interactions are omitted in the diagrams that are shown.
- It is assumed that by the time the file transfer begins, both the sender and the recipient have exchanged their capabilities using an OPTIONS or Presence exchange. Note that if there is a UX flow that does not show this, the assumption is that the OPTIONS or Presence requests were exchanged between the sender and the receiver (bidirectional) prior to starting the flow.

The RCS client shall populate the P-Preferred-Service header field in all CPM requests with the CPM Feature tag defined for the service, as described in [RCS5-CPM-CONVFUNC-ENDORS]. The S-CSCF or AS that performs the service assertion in the originating network shall add the P-Asserted-Service header field set to the value of the asserted CPM service ICSI (i.e. *urn:urn-7:3gpp-service.ims.icsi.oma.cpm.filetransfer* for CPM File Transfer or *urn:urn-7:3gpp-service.ims.icsi.oma.cpm.deferred* for deferred delivery done as part of the store and forward realization) and remove the P-Preferred-Service header field before further routing the request.

A receiving network element and RCS client should ignore any SIP header fields that they do not understand (e.g. P-Preferred-Service or P-Asserted-Service header fields).

3.5.4.1 File Transfer outside Group Chat

3.5.4.1.1 Selecting the file transfer recipient(s)

The user willing to share a file from the media gallery or file browser will select the file and choose the user with whom the file will be shared. The list that is presented initially to the

user may contain RCS contacts not currently registered and to which no store and forward is available. In addition, the capabilities the client has for a contact may not have been updated.

Therefore, the first step is to determine whether the file can be shared with the selected user (that is that user should be registered or be able to receive files using store and forward and the right capabilities should be in place).

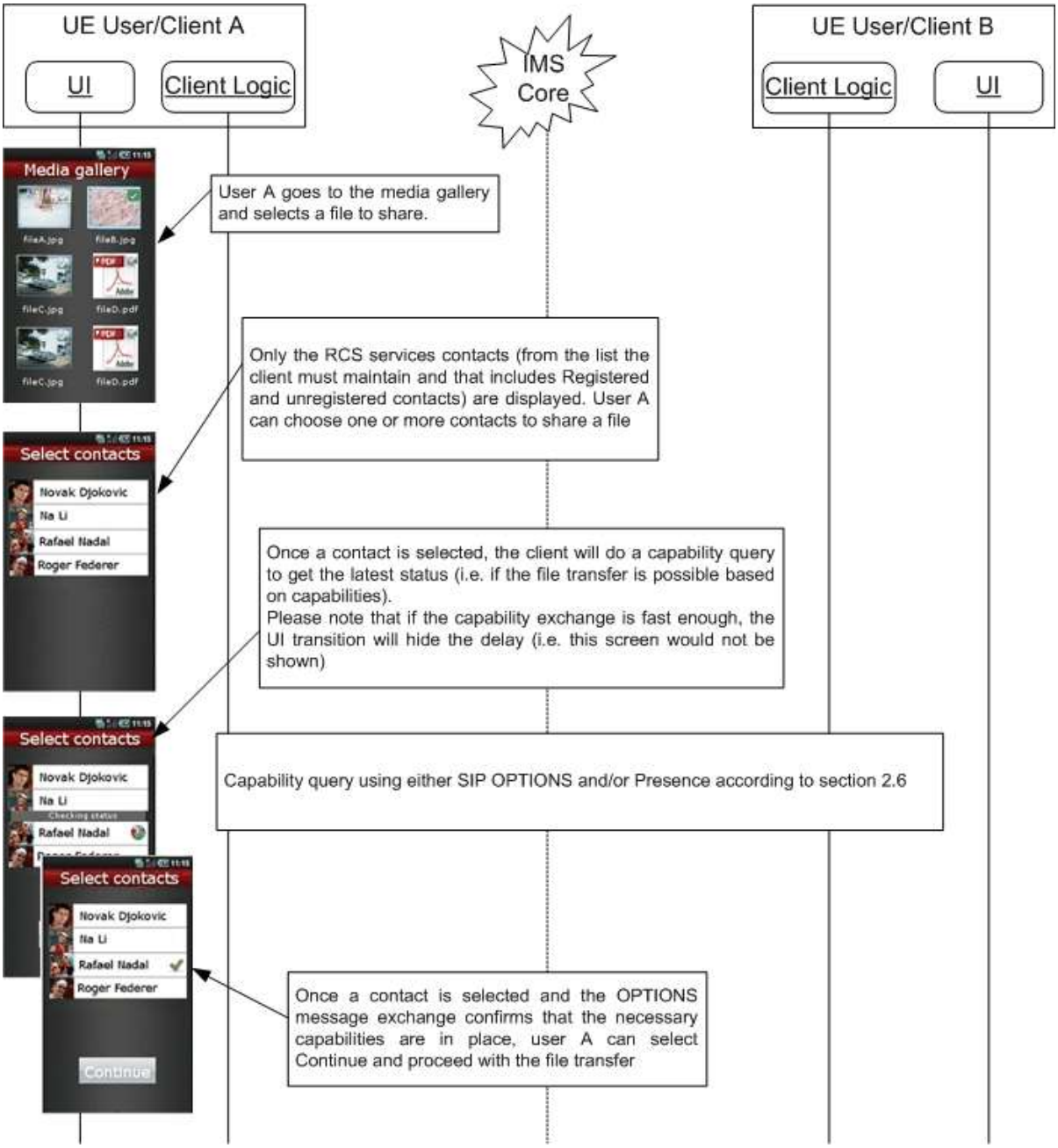


Figure 63: Selecting users when sharing a file from the media gallery/file browser

3.5.4.1.2 Standard file share procedure

Independently of the file share UX entry point, once the file and recipient are selected, the transfer can begin. If a user chooses to share several files, the individual file transfers (in each transfer only a single file is shared) are serialised by waiting for a SIP BYE before issuing the SIP INVITE request for the next file to transfer.

In the following diagram, it is assumed the receiver accepts the transfer.

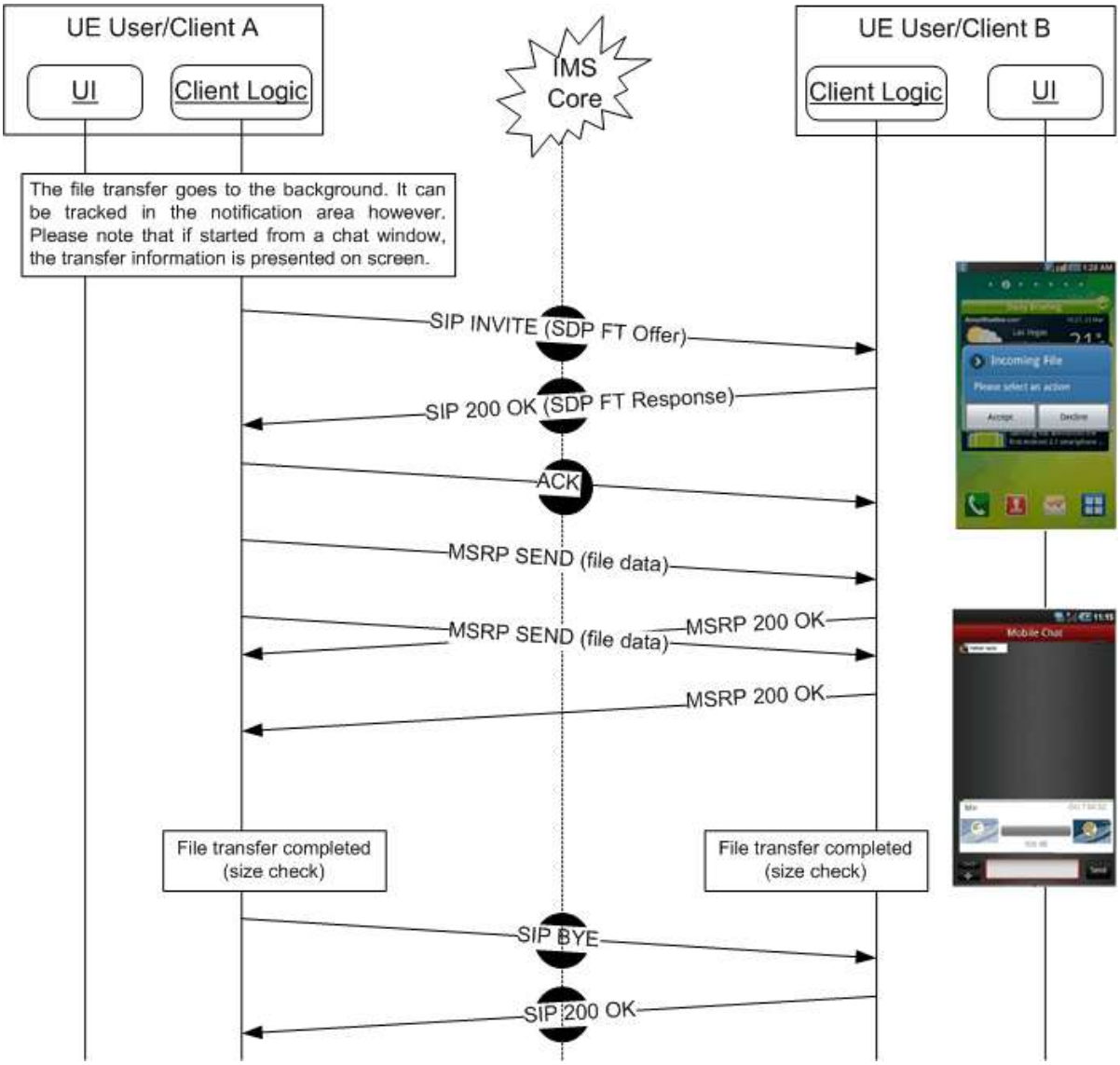


Figure 64: Standard file transfer sequence diagram – Successful transfer

As shown in Figure 64, a client shall send the file in different MSRP chunks without waiting on the MSRP 200 OK response before transmitting the next chunk.

As also shown in Figure 64, for a successful file transfer the client shall only send a SIP BYE after an MSRP 200 OK response has been received to all chunks of the file.

In the following diagram, User B rejects the transfer.

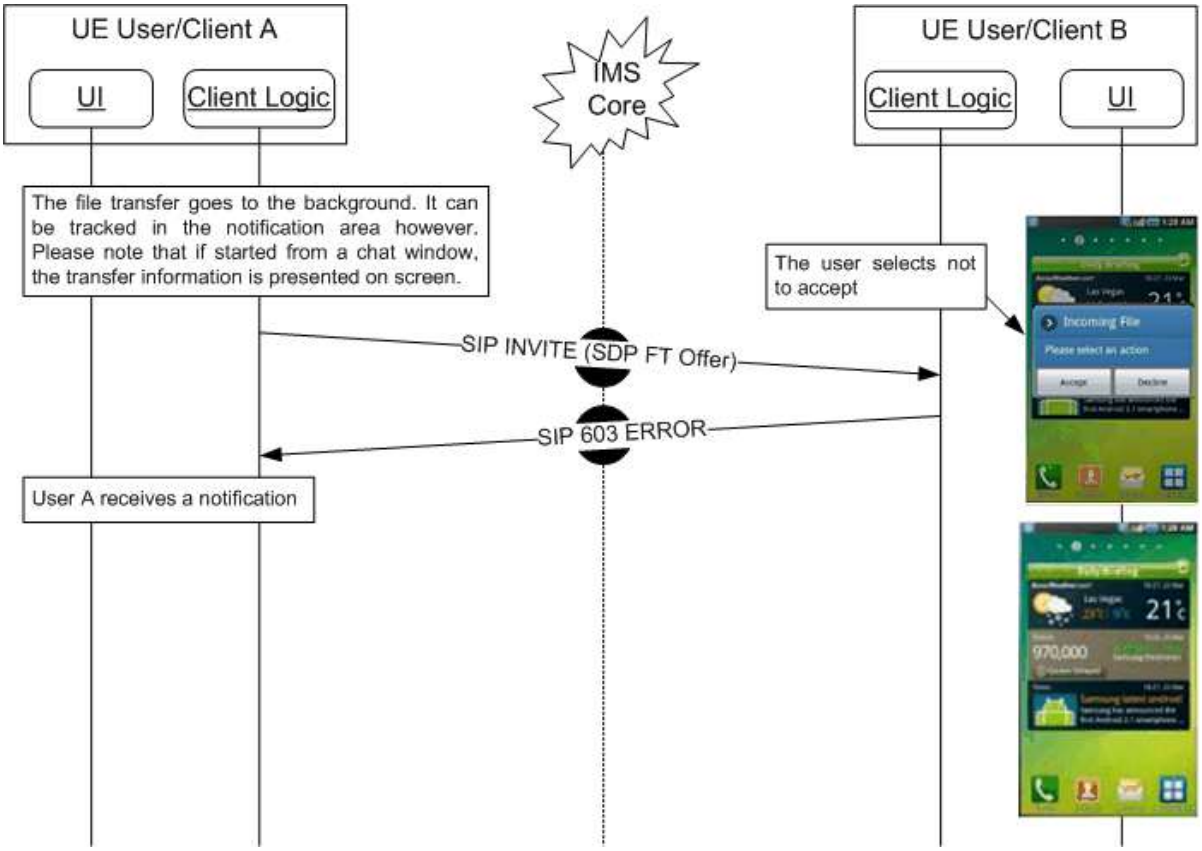


Figure 65: Standard file transfer sequence diagram – Receiver rejects the transfer

3.5.4.2 File Transfer in Group Chat

For File Transfer in Group Chat the file shall be sent to the conference focus. To support this the conference focus shall indicate in the Contact header field during the setup of the Group Chat whether File Transfer can be used in the Group Chat by including the IARI tags for the RCS File Transfer services it supports (see Table 23 and Table 28):

- The File Transfer Service itself
- The File Transfer Thumbnail
- File Transfer Store and Forward
- Geolocation PUSH (see section 3.10)

  NOTE1: These shall be included next to the ICSI for CPM Session Mode messaging or the *+g.oma.sip-im* feature tag if the chat is based on SIMPLE IM

Similarly, a client initiating, invited to or joining a Group Chat shall include those same tags in the Contact header it includes in respectively the SIP INVITE and 200 OK response for that Group Chat. To indicate its support for this mechanism, a client which is not capable of File Transfer at all shall include the tag for Chat defined in Table 23. Clients supporting File Transfer may include this attribute as well.

Also a Messaging Server accepting the Group Chat session on behalf of the user in Group Chat store and forward scenarios shall provide its capabilities in the Contact header allowing it to indicate for example whether for that user File Transfer is possible depending on store and forward for File Transfer being supported. When the Messaging Server takes the Group Chat session over from the user (e.g. when the user loses connectivity) or the user takes over from the Messaging Server (e.g. after regaining connectivity), the newly

applicable set of capabilities shall be announced to the conference focus using a session refresh (i.e. a re-INVITE with an updated Contact header) sent by the Participating Function.

When the conference focus indicated support for File Transfer, a client that wants to initiate a File Transfer shall compose a multiparty File Transfer Invitation as described in section 10.1 of [RCS5-SIMPLEIM-ENDORS] or section 7.4.1 of [RCS5-CPM-CONVFUNC-ENDORS] with following differences:

- The invitation shall be targeted at the IM Session Identity associated to the chat and thus to the conference focus instead of to the conference factory.
- The invitation shall take into account the capabilities of the focus. The client shall for example not include a thumbnail if not supported by the conference focus.
- No recipient-list shall be included.

  NOTE2: when File Transfer is not supported by the focus, a client can still send a file by initiating individual 1-to-1 File Transfer sessions to the chat participants

The participating functions and IMS will route the request to the focus that will generate individual INVITE requests for the participants as described in section 10.4 of [RCS5-SIMPLEIM-ENDORS] or section 9.3 of [RCS5-CPM-CONVFUNC-ENDORS] with following differences:

- The conference focus shall generate INVITE requests for all clients that indicated support for File Transfer in the Contact Header during the setup of the Group Chat. In the generated INVITE requests the conference focus will take into account the capabilities of the contact as indicated in that Contact Header and for example remove a thumbnail if one was included and no support for the thumbnail was indicated by the recipient
- The conference focus shall not generate INVITE requests to any recipients that has indicated not to support File Transfer in Group Chat (i.e. only the IARI for Chat was included in the Contact header for that participant)

  NOTE3: The conference focus may, based on local server policy, inform the participants in the chat that have indicated not to support File Transfer through a system message of the fact that a file transfer took place that they could not support.

- The conference focus shall handle participants that have not indicated capabilities in the Contact header during the setup of the Group Chat using one of the following options based on local server policy:
  - Not generate any INVITE requests for File Transfer for that participant (i.e. handle them in the same way as participants that have indicated not to support File Transfer in the Chat)
  - Generate a 1-to-1 INVITE request for a File Transfer on behalf of the initiator of the File Transfer without including a Thumbnail or supporting File Transfer store and Forward. If the recipient does not support File Transfer this may fail.
- For those participants that have announced their capabilities through the Contact header, the conference focus may target the INVITE request at only the device of the participant that is handling the chat session through the mechanisms detailed in section 2.11.3.
- The conference focus shall not include a recipient-list-history body in the generated INVITE requests
- The conference focus may, based on local server policy, limit the number of ongoing file transfers to a participant.

A client can detect that a File Transfer request coming from the conference focus is associated with a Group Chat in which it is involved this because the File Transfer Contact Header matches the one from the associated Group Chat.

If the Group Chat has been closed due to inactivity when the user wants to send the file to the chat, the Group Chat will be reactivated as stated in section 3.4.4.1.7 before sending the file. After the delay to allow this reestablishment as described in the same section, the File Transfer will be initiated to the Group Chat's focus.

As in the case of store and forward (See sections 3.5.4.7 and 3.5.4.8) client shall use a CPIM wrapper to request delivery reports if the user wants to be informed about the delivery of the file to the different participants. This wrapper shall be handled by the conference focus for those recipients that do not support store and forward (as indicated in the Contact header provided during the setup of the Group Chat session). The conference focus shall send the content to such a participant without CPIM wrapper and when the file is delivered to that participant the focus may send the corresponding delivery report to the sender either through a SIP MESSAGE (similar to the case where it is generated by the participating function or through an MSRP SEND request in the Chat session.

Note that contrary to the Group Chat itself, for a File Transfer the conference focus has to provide all packets sent in the session to a participant that is late to accept rather than just those that are sent from the moment the recipient joins.

If the initiator of the File Transfer loses connectivity during the transfer, the initiator may attempt a resume of the File Transfer after re-joining the Group Chat. If connectivity is lost by a recipient, that recipient may attempt a resume after rejoining the Group Chat (as described for File Transfer Store and Forward in section 3.5.4.7). That resume request won't be relayed by the Group Chat focus to the initiator of the corresponding File Transfer though. Instead the focus shall send a SIP *488 Not Acceptable Here* Error Response.

If the Group Chat is terminated while a File Transfer is ongoing, the Group Chat conference focus may, based on local server policy, interrupt the ongoing File Transfer. Whether this is done can depend on the reason the Group Chat session was terminated.

All this leads to the following flow:

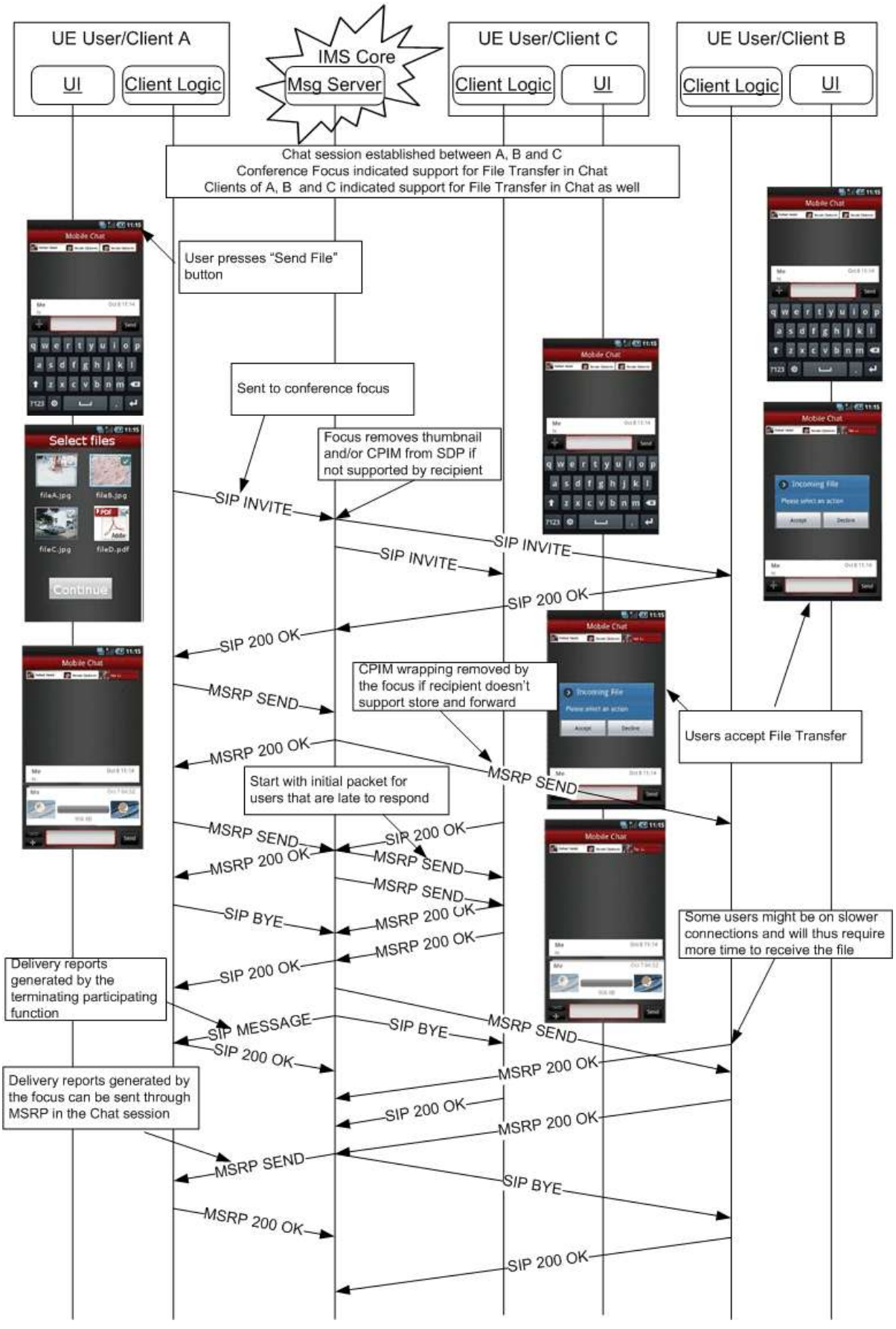


Figure 66: File Transfer in Group Chat

3.5.4.3 File share error cases

There are several scenarios in which a file transfer can result in an error:

Either the sender or the receiver decides to cancel the operation before the transfer is completed. The relevant sequences are equivalent to the diagrams presented for image sharing during a voice call in sections 3.6.4.3.8 and 3.6.4.3.9.

NOTE1: Because in RCS only a single file is transferred in a file transfer session this simplified procedure for the receiver shall be used instead of the one defined in [RFC5547] (referred to from [RCS5-SIMPLEIM-ENDORS] and [RCS5-CPM-CONVFUNC-ENDORS]).

Either the sender or the receiver loses the connection to the network before the transfer is completed. The relevant sequences are equivalent to those presented for image sharing during a voice call in sections 3.6.4.3.12 and 3.6.4.3.13.

When transferring larger files, the probability is higher that such a transfer would be interrupted. If such an interrupt leads to termination of the underlying MSRP session, the receiving client, knowing the overall size of the file in transfer, will become a requester of a file (as described in [RFC5547]) and sends a SIP INVITE request, specifying in the SDP payload this file (by using the file-selector as described in [RFC5547]) and the missing part of the file, using the file-range attribute.

Finally, note that if during a file transfer the capabilities of one of the ends change, the file transfer may be affected:

- If the receiver runs out of storage space, the sequence should be equivalent to that presented in section 3.6.4.3.10.
- If on one of the ends a handover into 2G (2G GPRS data coverage) occurs without losing the IP configuration, the file transfer should continue until finished.

If the PNB feature is supported[27], the BPEF checks the recipient of a file transfer against the originating PNB '*rcs_pnb_outft_blockedusers*' of the sender. If the recipient is found on the list, the BPEF will reject the setup of the SIP INVITE session with the blocked user.

NOTE2: For File Transfer, the BPEF may be located in the Messaging Server.

3.5.4.4 File share and file types

In principle the RCS file transfer service comes without a limitation on the file sizes or types. This means that any kind of file can be transferred using this service. Taking this into account and with the aim of providing all the necessary facts to the receiver allowing making an informed decision on whether to accept or to reject the file, a user receiving a file transfer invitation should be informed at a minimum of:

- The size of the file: This is mainly to protect the user from unexpected charges and/or long transfers.

  NOTE: this also applies to the sender.

- The file type: In this case and to make it more intuitive, the device should present to the user whether the file which is being transferred can be handled/displayed by the device.

For example, if a user receives an invitation to receive a PDF (Portable Document Format) document and their device cannot process that document, an informative message with the

[27] The present section assumes the BPEF as described in section 2.15.1 is provided by the Messaging Server.

size and the fact that the file type is not supported should be presented to the user prior to the user making the decision on accepting or rejecting the file transfer.

Finally note that each individual Service Provider may introduce restrictions taking into account different considerations such as security, intellectual property and so on.

3.5.4.5 Personal Network Blacklists handling

NOTE: In the present section, the BPEF as described in section 2.15.1 may be provided by the Messaging Server.

The Personal Blacklists are applied by the BPEF at both origination and termination of file transfer.

The following *resource-lists* from Shared XDMS are checked by the BPEF by comparing the URI values used in the request and in the list:

- at origination:
  a) the BPEF of the originator checks the '*rcs_pnb_outft_blockedusers*' list to verify that the recipient is not among the blocked users for this request by comparing URIs contained in the list with the URI value of the Request URI of the SIP request.
  b) if this is the case, the message is discarded and a SIP a *403 Forbidden* response with a warning header set to "122 Function not allowed" is returned to the user.
- at termination:
  a) The BPEF checks the '*rcs_pnb_ft_blockedusers*' list, to verify if the originator of the file transfer is among the blacklisted users by comparing the URIs contained in the list with the URI values of the *P-Asserted-Identity* header field of the SIP request.
  b) If true, the BPEF returns a *403 Forbidden* with a warning header set to "122 Function not allowed"
  c) If the Common Message Store is supported it shall store the File Transfer History object data as defined in [RCS5-CPM-MSGSTOR-ENDORS] for the blocked File Transfer event in user's dedicated Blocked Folder.

3.5.4.6 File size considerations

To prevent both the abuse of the file transfer functionality and protect customers from unexpected charges, a configurable size limitation (refer to FT WARN SIZE and FT MAX SIZE in Table 86 for reference) may be enabled.

From the user experience perspective and assuming that the size limitation is in place (i.e. the values are non-zero):

- If a file transfer (send or receive) involves a file bigger than FT WARN SIZE, the terminal should warn the user of the potential associated charges and get confirmation from the user to proceed.
- If the file is bigger than FT MAX SIZE, a warning message is displayed when trying to send or receive a file larger than the mentioned limit and the transfer will be cancelled (that is at protocol level, the SIP INVITE request will never be sent or an automatic rejection response SIP 403 Forbidden with a Warning header set to "133 Size exceeded" will be sent by the entity that detects that the file size is too big to the other end depending on the scenario).

3.5.4.7 File transfer store and forward using a MSRP-based File Transfer Function (FTF)

This functionality requires a logic server function identified as the File Transfer Function (FTF).

NOTE: As a logical function this can be either provided as part of a physical application server that it is already providing analogous functionality (e.g. Messaging AS) or in a separate one.

This procedure allows the file store and forward mechanism for the following use cases:

1. When the receiver ignores the file transfer invitation causing the SIP INVITE procedure to expire or an early expiration due to an error.
2. When the receiving user is offline
3. When the either sender or receiver loses connectivity

This is reflected in Table 56 that shows the error responses that will result in the FTF storing the file:

Response received on terminating leg	Response sent on originating leg	Store the file
480 Temporarily unavailable	200 OK	Y
408 Request Timeout	200 OK	Y
487 Request Terminated	200 OK	Y
500 Server Internal Error	200 OK	Y
503 Service Unavailable	200 OK	Y
504 Server Timeout	200 OK	Y
600 Busy Everywhere	200 OK	Y
Any other response (including 404 Not Found, 603 Decline, 403 Forbidden and 200 OK)	Received response (that is no mapping is done)	N

Table 56: Mapping of received Error Responses by the FTF

3.5.4.7.1 File transfer invitation

If supported by a service provider, this functionality shall be provided by the terminating side (receiver) and, optionally, it can be also provided by the originating side, as per the steps provided below:

1. After the capability exchange takes place, the sending client shall verify that both the sender and the receiver support the file transfer store and forward feature.
2. The original file transfer SIP INVITE shall include a CPIM/IMDN body requesting a delivery notification as described in section 3.3. Note that in this case no message is carried in the CPIM body. This allows the sender to request a delivery notification to confirm when the receiver gets the file.
3. After the SIP INVITE is sent towards the receiver, the terminating FTF shall intercept the message before it is forwarded to the receiver (via normal IMS initial filter criteria already in place for RCS features) and store the file-transfer-id and the file-name SDP attributes defined in [RFC5547].
4. From this moment the terminating FTF shall forward the INVITE to the destination client, and can give a chance for the destination client to accept the File Transfer by waiting for a SIP response during a configured period of time.
   a) If the destination client accepts or refuses the file transfer, before the end of this configured period of time, the standard procedures apply with the precision given in step 5.
   b) Otherwise, after the expiration of the corresponding configured timer, the terminating FTF shall cancel the SIP INVITE request towards the receiver by sending a SIP

CANCEL. To make sure the receiver client understands that the reason for this cancellation is a timeout, a reason header shall be included as presented in the following table consistently with [RFC3326]:

Reason: SIP;cause=408;text="User not responding"

Table 57: Reason header in SIP CANCEL due to timeout

c) If an error occurs that is listed in Table 56, the FTF shall also accept the file transfer on behalf of the destination user and store it.

Please note that specifying when the FTF should accept the initial SIP INVITE and start storing the file transfer is outside the scope of this UNI specification and it is left up to Service Provider policy. Possible implementation choices are:

- Accept the file transfer when the CANCEL is sent to the end user or an error is received (note that this is the option shown in the figures below).
- Accept the file transfer as soon as the initial INVITE has been received.

If the transmission is interrupted from the sender (e.g. because of loss of connectivity), it is left up to the local policy of the FTF whether the received fragment remains stored and if so for what time or whether it is discarded. In case it is discarded (either immediately or after expiry) and the sender tries to resume the file transfer, the file transfer will be rejected as described in [RFC5547]. In that case the sender shall transmit the entire file again as described in section 3.5.4. A stored fragment of a File Transfer shall not be forwarded to the recipient until the sender has resumed the File Transfer and provided the remainder of the file.

5. When the destination client accepts the file transfer invitation by sending a SIP “200 OK”, the terminating FTF should always stay in the media path to be able to have a local copy of the file. How the local copy is performed is outside the scope of this specification and is up to each service provider. If the recipient client loses connectivity, the terminating FTF should complete the File Transfer on the incoming leg anyway in order to be able to, later, provide the file to any potential resume request from the recipient as per procedure defined in section 3.5.4.
   The local copy is only needed until the file is received by the recipient. The FTF shall delete the local copy, once the file has been completely received by the recipient.

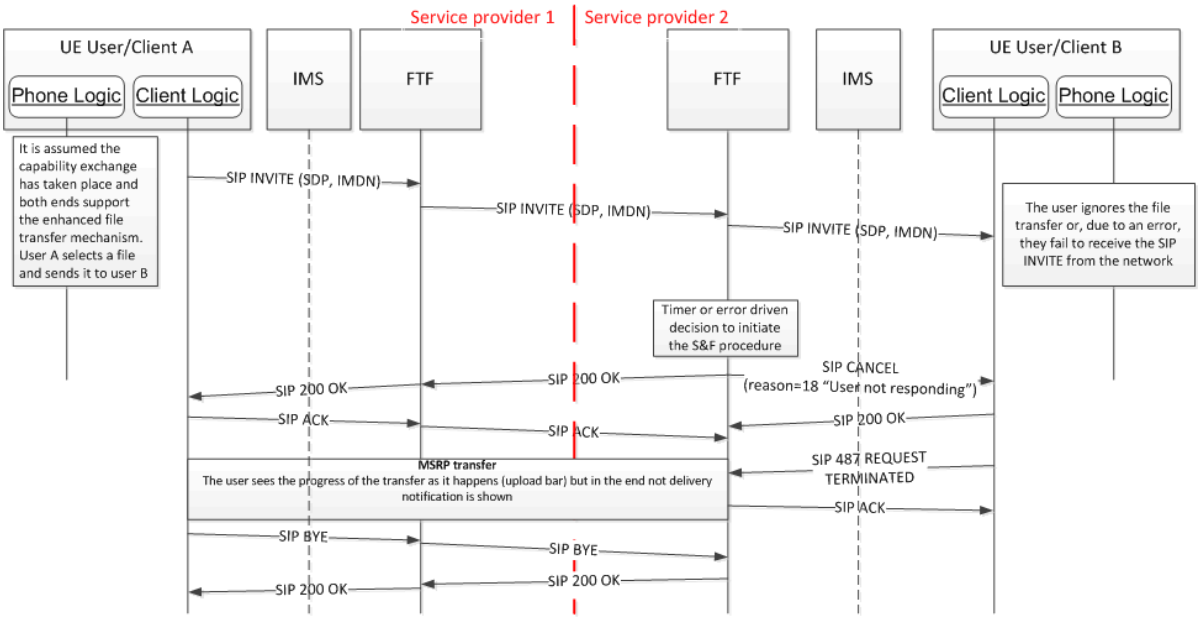


Figure 67: File transfer with store and forward via MSRP fetch on terminating service provider

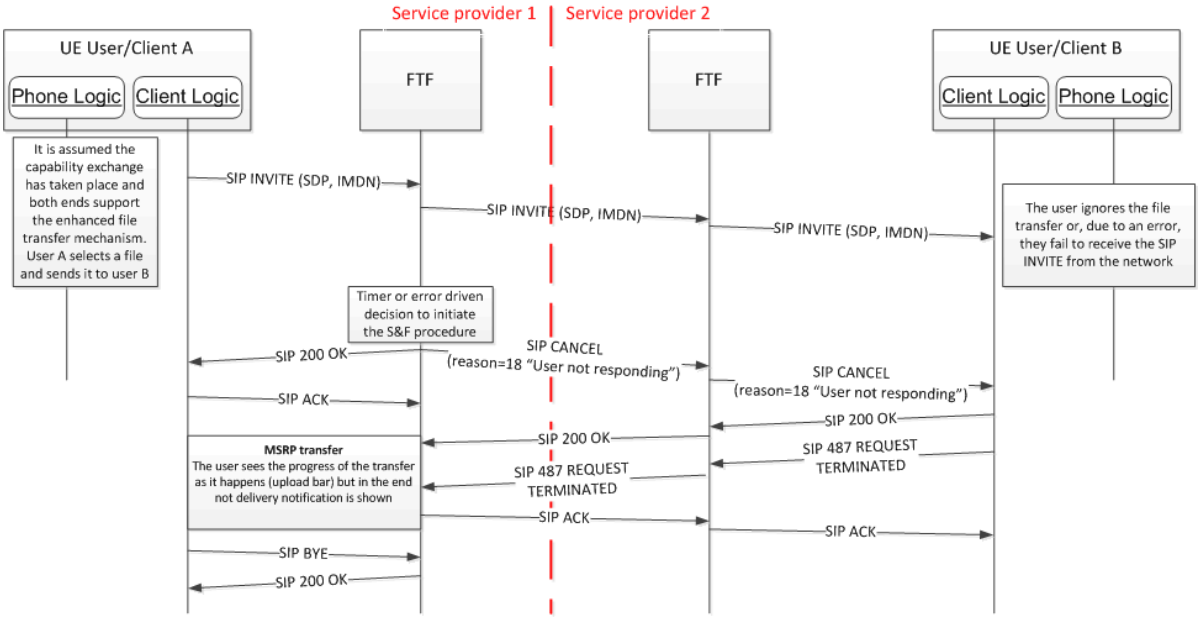


Figure 68: File transfer with store and forward via MSRP fetch on originating service provider

Please note the fetching procedure is covered in section 3.5.4.7.3.

3.5.4.7.2 File transfer to offline users

If supported by a service provider, files can be sent to users that are not online. This functionality can be provided by either the originating (senders) or terminating (receiver's) service provider.

On the originating client this can be enabled by having the *FT CAP ALWAYS ON* parameter (defined in Table 86 in section A.1.4) set to 1, indicating that the file transfer can take place even if the recipient is offline. This parameter should only be set to 1 if either:

1. All the interconnected service providers support the file transfer store and forward feature, or,
2. Store and forward for files is provided as an originating function (sender's FTF).

In this case File Transfer shall be offered towards all users that are known to support the File Transfer service based on a prior capability exchange.

Also when *FT CAP ALWAYS ON* is set to 0, the originating client may, based on a prior capability exchange, as described in section 2.6 be aware that the recipient that is offline supports Store and Forward for File Transfer. If *FT CAP ALWAYS ON* is set to 0 File Transfer shall not be offered to offline recipients that are not known to support Store and Forward for File Transfer.

When initiating a File Transfer to an offline user, the client shall compose the file transfer SIP INVITE request as described in [RCS5-SIMPLEIM-ENDORS] or [RCS5-CPM-CONVFUNC-ENDORS] (depending on the used message technology) with following changes:

1. It shall include a CPIM/IMDN body identical to that described for the chat/IM service in section 3.3.4.1 except that in this case a display notification is not requested and no message is carried.
2. It shall not include the FT thumbnail, since it is not known if the recipient or the recipient's network has this capability.

There are two possible cases:

1. The receiver's Service Provider supports the RCS File Transfer store and forward procedures and is aware that the receiver is offline or receives a SIP 408 Request Timeout or SIP 480 Temporary unavailable error when sending the request to the client, and therefore accepts the file transfer on their behalf.
    - When the receiver's FTF has detected that the receiver is back online (i.e. third party registration) the FTF forwards the SIP INVITE request without the CPIM/IMDN body. In order to support legacy devices, the file transfer SIP INVITE request shall carry the P-Asserted-Identity of the original sender, rather than the identity of the FTF that stored the message.
    - The receiver's FTF will take the responsibility to issue the delivery notification back to the originator.
2. The sender's Service Provider supports the RCS File Transfer store and forward procedures.
    - In this case, the FTF may not be able to detect when the user comes back online, and must therefore periodically retry to send the File Transfer SIP INVITE request to the recipient. The retry period and duration is determined by local Service Provider policy (see section 3.5.4.7.2.1).

From this point on, the standard file transfer procedure and the cases covered in the remaining sub-sections of section 3.5.4.7 apply.

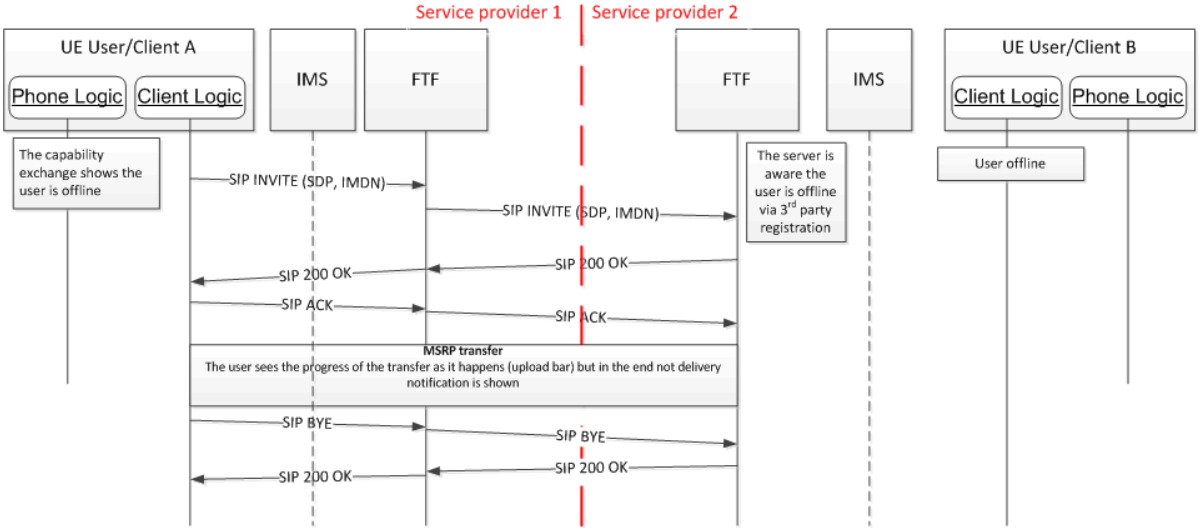


Figure 69: File transfer with store and forward via MSRP for offline users (1/2)

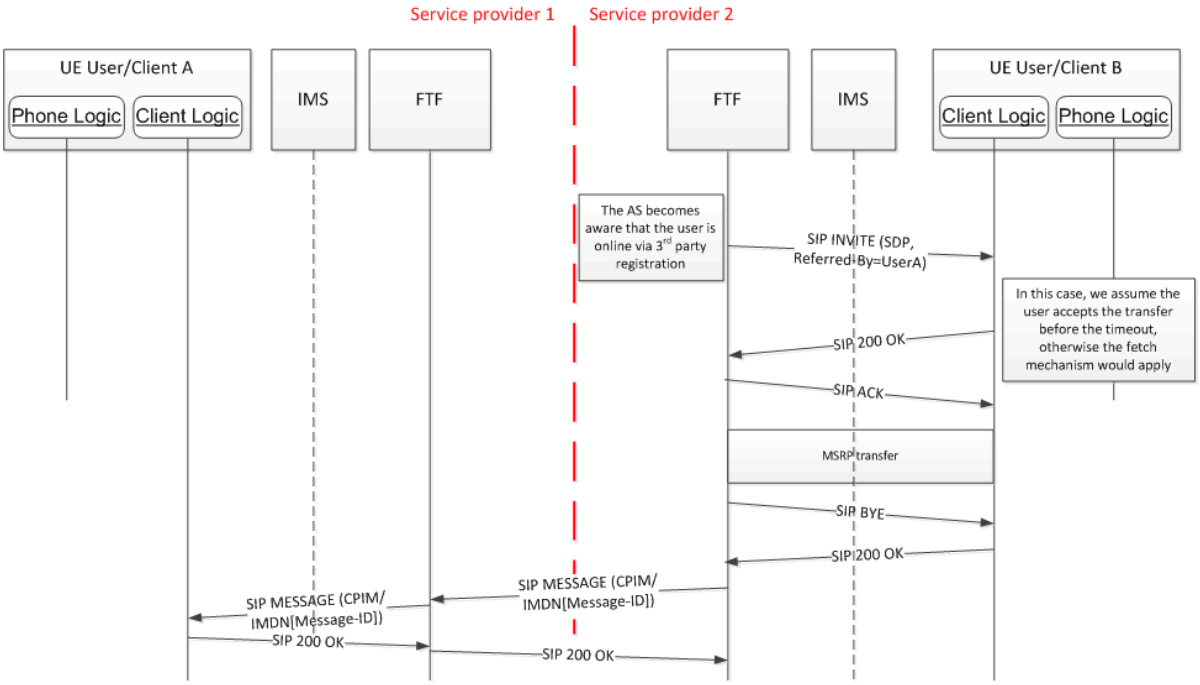


Figure 70: File transfer with store and forward via MSRP for offline users (2/2)

3.5.4.7.2.1 File Transfer retries in originating network

If the sender's network provides store and forward functionality, the sender's FTF will accept the File Transfer request if one of error responses listed in Table 56 is returned from the terminating network.

In this case the originating FTF shall attempt retries to deliver the file towards the receiver. The following procedure will handle both legacy and RCS 5.1/5.2 receiving devices.

1. A normal file transfer SIP INVITE request is sent from the sender's FTF to User B as described in [RCS5-SIMPLEIM-ENDORS] or [RCS5-CPM-CONVFUNC-ENDORS] (depending on the used message technology) with following changes:
   - The SIP INVITE contains file transfer feature tag and the identity of the original sender in the *P-Asserted-Identity* header.

- This SIP INVITE shall be sent without the CPIM/IMDN body containing the delivery notification request (i.e. like in the case of a file transfer without the store and forward functionality).

2. When the receiver's device is online and the user accepts the File Transfer, the file shall be transferred.
3. When the file is delivered, the FTF shall issue the delivery notification back to the originator and should delete the stored copy of the file.
4. If the receiver's device is not available, a *480 Temporary Unavailable* error can be expected. If that or another error listed in Table 56 occurs:
   - The sender's FTF re-tries with step 1 after a Service Provider configurable amount of time.
5. If the Service Provider defined number of retries or amount of time has elapsed or a SIP *603 Decline* response is received, the undelivered files are discarded, and the sender is notified if requested.

3.5.4.7.3 Client behaviour and file fetching

After receiving the cancellation (protocol-cause 408 in the Reason header field indicating that store and forward took place), the RCS client shall try to fetch the file as presented below:

1. The receiver client/device implementation, knowing that the original SIP INVITE is expired, shall fetch the file from the FTF. In order to identify the requested file uniquely the client shall:
   a) Use the same SDP file-transfer-id that was used in the original SIP INVITE
   b) Use the same SDP file-name that was used in the original SIP INVITE.

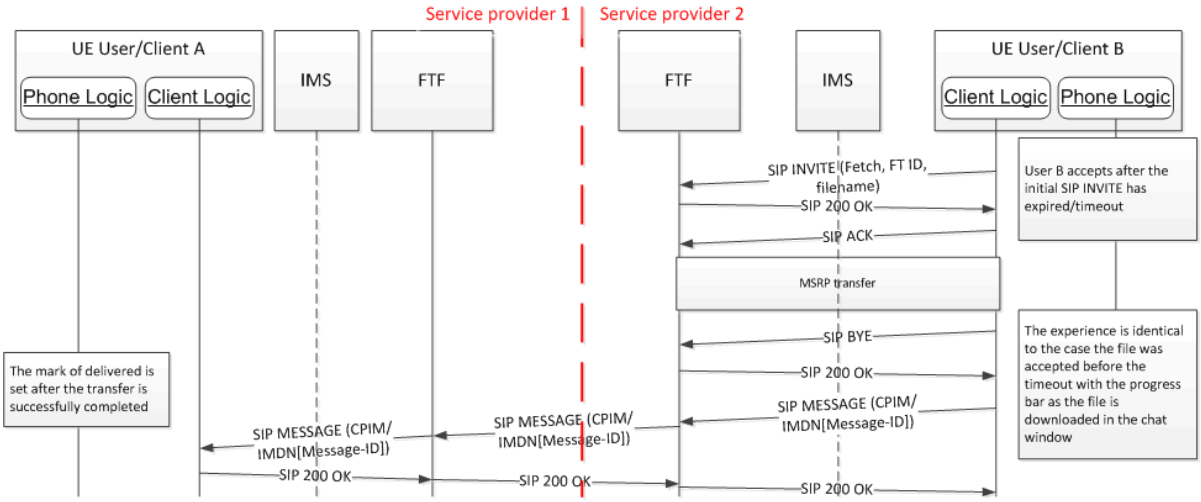


Figure 71: File transfer store and forward via MSRP on terminating FTF fetch

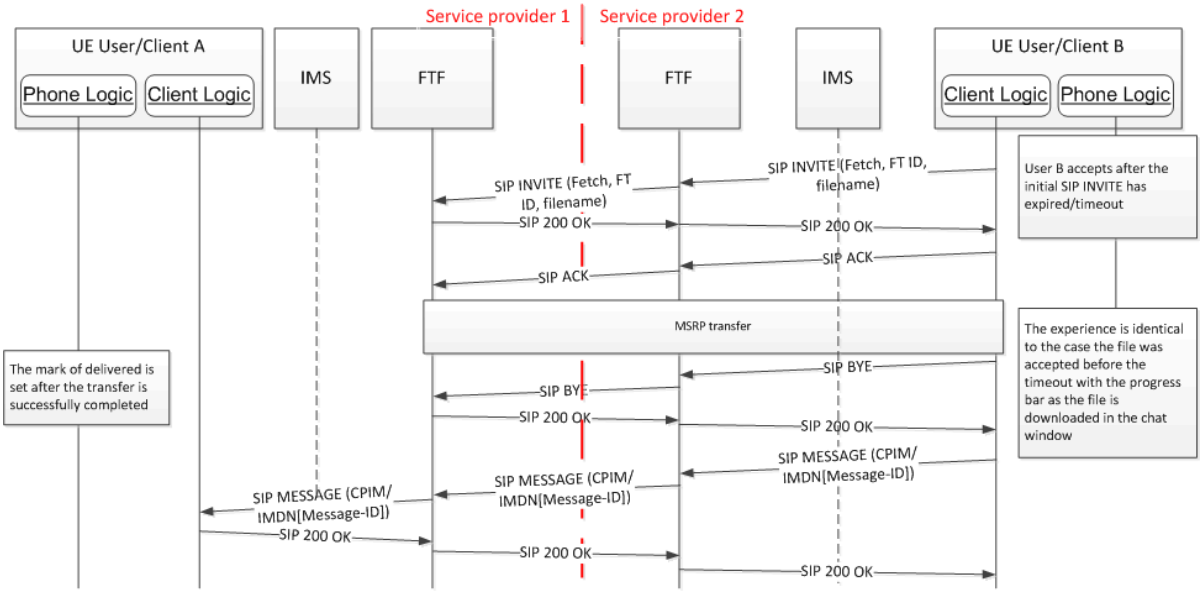


Figure 72: File transfer store and forward via MSRP on originating FTF fetch

2. The server will remove the file once it is successfully downloaded.
3. After the file is successfully downloaded a SIP MESSAGE containing the delivered notification will be issued to the sender to confirm the destination got the file.

3.5.4.7.4 Timing between originating and terminating store and forward

When implementing store and forward, the timing to trigger the store and forward procedure shall take into account whether store and forward is supported on the terminating side, the timer (or time to trigger the error that signals store and forward is required) shall be significantly smaller than the timer used on the originating store and forward process. Consequently, the following recommendations shall be followed:

- The timer on the originating side should be greater than ½ the SIP INVITE timeout period
- The timer on the terminating side (or time to trigger the error) should be smaller than ¼ the SIP INVITE timeout period

3.5.4.7.5 File transfer procedures without store and forward

Following the capability exchange and assuming both sender and receiver support the store and forward procedures, there are two possible scenarios where the file transfer procedure does not require a store and forward:

1. If the receiver accepts before the SIP INVITE expiration, then the file transfer takes place as normal:
   a) The MSRP session is established to perform the file transfer.
   b) When completed a SIP BYE is exchanged to terminate the session
   c) Please note that the original file SIP INVITE is modified to include a CPIM/IMDN body identical to that described for the chat/IM service in section 3.3 except for the fact that in this case a message is not carried and a display notification is never requested. This allows the sender to request a delivery notification to confirm when the receiver gets the file. In this case, as no store and forward takes place, the receiver client is responsible to issue a SIP MESSAGE containing the CPIM/IMDN notification that the file has been successfully delivered.

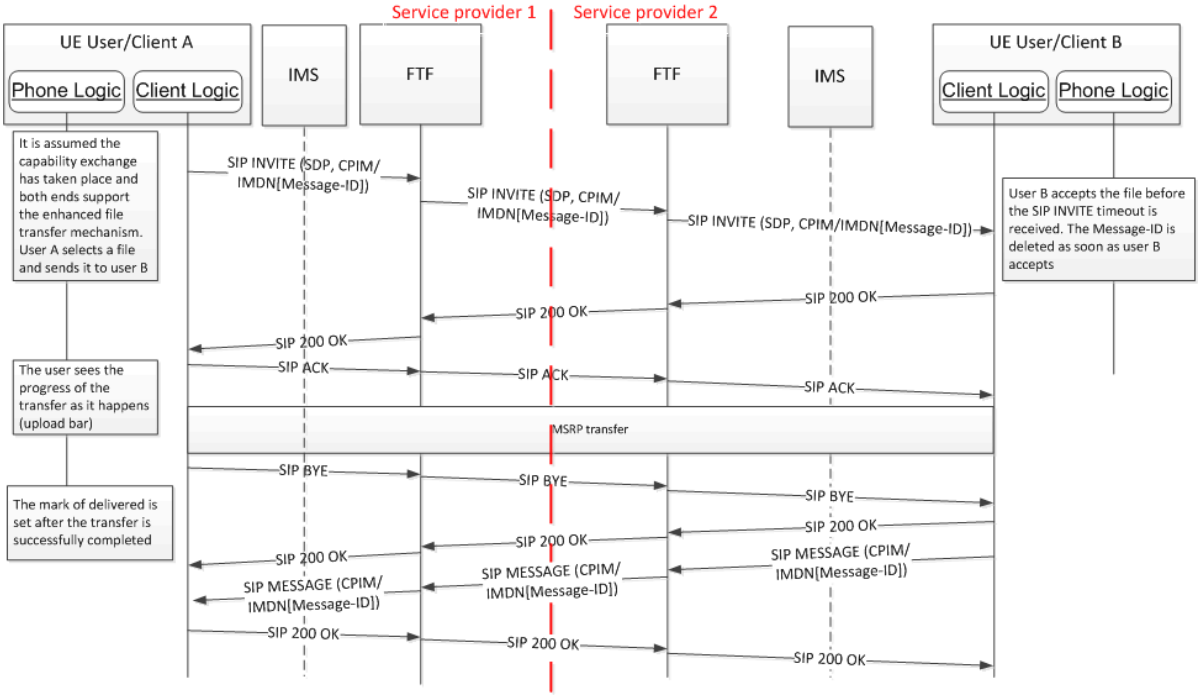


Figure 73: File transfer without store and forward: Receiver accepts file before timeout

2. If the receiver rejects before the SIP INVITE expiration, then a 603 DECLINE response is sent to the sender and the file transfer is cancelled.

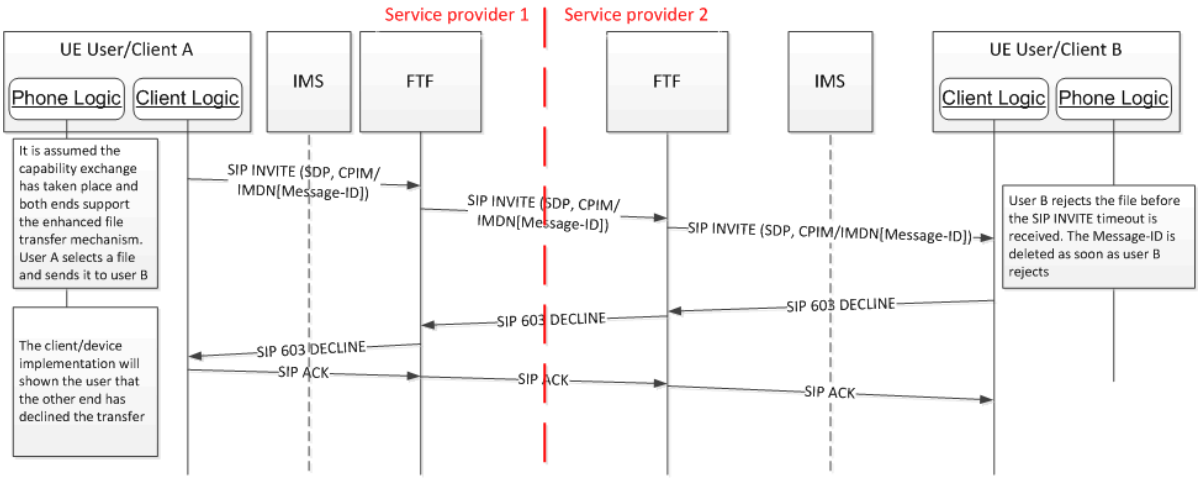


Figure 74: File transfer without store and forward: Receiver rejects file before timeout

3.5.4.7.6 CPIM/IMDN delivery notifications

Note that the same mechanism used for the 1-to-1 chat described in section 3.3.4 specification shall be used with the following changes:

- The notifications shall always be sent using a SIP MESSAGE
- The IMDN disposition shall ONLY include a delivery notification (request or response depending on the case) and never request or generate a displayed notification

3.5.4.8 File Transfer via HTTP

As presented in the previous sections, the default mechanism to transfer files in RCS is based in a MSRP transfer.

The present section proposes an alternative mechanism where the file transfer is based in storing the file in a publicly available server and then sharing the location using standalone messaging and the 1-to-1 and Group Chat procedures described in sections 3.2, 3.3 and 3.4. Message revocation procedures as described in section 3.3.4.1.10, do not apply for 1-to-1 Chat messages carrying the location where the file is stored. The same applies for all services that utilise File Transfer via HTTP mechanism (e.g. audio messaging). The main motivations behind this procedure presented below:

- Through the reuse of the procedures for RCS messaging (standalone messaging and chat), the HTTP file transfer mechanism shall automatically benefit from the store and forward mechanism available for chat meaning there is no need to specify additional store and forward procedures
- HTTP is a quite mature protocol broadly supported for many years in the majority of terminals. This procedure shall therefore benefit from its availability and resiliency.

3.5.4.8.1 Configuration and capability exchange

In order to guarantee back compatibility the file transfer via HTTP procedure shall be only used instead the MSRP procedure if:

1. The sender is adequately configured to use this procedure which is verified by checking that the *FT HTTP CS URI*, *FT HTTP CS USER* and *FT HTTP CS PWD* configuration parameters (all defined in Table 86 in section A.1.4) are present and correctly set in the configuration received by the file sending client.
2. Both sender and receiver support the procedure by verifying that the File Transfer via HTTP capability defined in section 2.6.1 is present in the RCS capabilities on both ends. Note that an RCS client shall only make this capability available if the service is supported by the implementation and the configuration parameters *FT HTTP CS URI*, *FT HTTP CS USER* and *FT HTTP CS PWD* are correctly set. In this case the RCS client shall also include the File Transfer via HTTP IARI tag defined in section 2.6.1.1.2 in the Contact header of the SIP INVITE requests and SIP 200 OK responses that it sends during the setup of a Group Chat.

If both ends support the procedure, all file transfer shall be performed using the new procedure described in this and the following sections. If not, the file transfer via MSRP procedures described 3.5.4 shall be employed.

Note that when both ends are in chat or group chat session the capability shall be available if following conditions are fulfilled:

- the application/vnd.gsma.rcs-ft-http+xml content type is indicated in the a=accept-wrapped-types attribute during the SDP negotiation and
- for the case of a 1-to-1 chat, the contact is known to support the File Transfer via HTTP capability based on a capability exchange or on the cached result of an earlier capability exchange when a capability exchange doesn't provide a conclusive result and
- For the case of a Group Chat, the Contact header field received during the setup of that Chat included the File Transfer via HTTP IARI tag defined in section 2.6.1.1.2.
  A conference focus supporting File Transfer via HTTP shall therefore indicate this support by including the File Transfer via HTTP IARI tag defined in section 2.6.1.1.2 in the Contact Header field of the SIP INVITE and SIP 200 OK responses that it sends during the setup of the Group Chat. When one of the participants in the Chat initiates a File Transfer via HTTP, the conference focus shall not forward the File Transfer via HTTP body to participants that did not include the File Transfer via HTTP IARI tag in the Contact header that they provided during the setup of the Group Chat.

3.5.4.8.2 Offline users

RCS client shall allow the file transfer via HTTP even the receiver is offline when:

- IM CAP ALWAYS ON configuration parameter (defined in sections A.1.3.3) is set to enabled (1), and,
- the receiver user is known to support the file transfer over HTTP capability (cached from the previous exchange).

3.5.4.8.3 File transfer procedure

3.5.4.8.3.1 Sender procedures

Note In this whole section it is assumed that the sender has the *FT DEFAULT MECH* configuration parameter (see section A.1.4) is set to HTTP.

1. After the capability exchange takes place, it is verified whether both the sender and the receiver support the file transfer via HTTP procedure (as described in section 3.5.4.8.1 and 3.5.4.8.2).
2. Assuming both ends support it, the sender shall first send an empty HTTP POST[28] request (i.e. a request without any body) to the FT HTTP CS URI. This request shall result in any of following responses
    a) a HTTP 401 AUTHENTICATION REQUIRED error response carrying a WWW-Authenticate header field as defined in [RFC2616] if authentication is required
    b) A.HTTP 204 NO CONTENT response if authentication is not required
    c) a HTTPS 503 INTERNAL ERROR with retry-after header if the server is busy and cannot handle the request. The RCS client shall in this case retry to upload after the time specified in the retry-after header.
    d) Any other response, the RCS client shall retry the request in this case.
3. The sender shall then upload the file to the HTTP content server by making a HTTPS POST request to the FT HTTP CS URI to upload the file containing the following three elements:
    - A file transfer Transaction ID (TID): this TID is optional and is included in case the client supports the optional resume of the file upload as described in section 3.5.4.8.3.1.1. The TID value shall be a unique ID generated by the client according to [RFC4122] section 4.2.
    - The thumbnail content: This is optional as it is only required for images and videos as per the procedures described in section 3.5.4
    - The file content

    If authentication to the server is required, it shall depending on the WWW-Authenticate header received in step 2 be performed using basic authentication or HTTP digest as per [RFC2617] using *FT HTTP CS USER* and *FT HTTP CS PWD* configuration parameters as credentials.

    In order to carry these three elements, the HTTP POST method shall contain a MIME *multipart/form-data* entity body with the following parts that shall be transmitted in the listed order:

[28] This specification uses the term “HTTP POST” and “HTTP GET” as a generic reference to the action of using the POST or GET methods of HTTP. However, it is strongly recommended that whenever the POST action contains sensitive information such as a user ID or password, the action should take place over a secure connection and/or via HTTPS explicitly. This is enforced by the service provider by configuring a FT HTTP CS URI with "https" schema.

- An optional one containing the transaction ID:

```
Content-Disposition: form-data; name="tid”
Content-Type: text/plain

<Transaction-ID generate by the client>
```

Table 58: First form of the HTTP POST method request to upload the file to the HTTP content server (Transaction ID)

- An optional one containing the thumbnail:

```
Content-Disposition: form-data; name="Thumbnail"; filename="<local_filename>"
Content-Type: [mime type depending on the thumbnail; e.g. image/jpeg]

<Thumbnail content>
```

Table 59: Second form of the HTTP POST method request to upload the file to the HTTP content server (Thumbnail contents)

- One containing the file:

```
Content-Disposition: form-data; name="File"; filename="<local_filename>"
Content-Type: [mime type depending on the file; e.g. image/jpeg]

<file content>
```

Table 60: Third form of the HTTP POST method request to upload the file to the HTTP content server (file contents)

4. There are two possible cases:
   a) If the upload is successful, the client shall get a HTTPS 200 OK response containing a XML in the body that specifies:
      i. The URL, size, content type and validity for the thumbnail, if applicable
      ii. The URL, size, filename, content type and validity for the file

```
<?xml version="1.0" encoding="UTF-8"?>
< file xmlns="urn:gsma:params:xml:ns:rcs:rcs:fthttp">
        <file-info type="thumbnail">
                <file-size>[thumbnail size in bytes]</file-size>
                <content-type>[MIME-type for thumbnail]</content-type>
                <data url = "[HTTP URL for the thumbnail]" until = "[validity of the thumbnail]"/>
        </file-info>
        <file-info type="file">
                <file-size>[file size in bytes]</file-size>
                <file-name>[original file name]</file-name>
                <content-type>[MIME-type for file]</content-type>
                <data url = "[HTTP URL for the file]" until = "[validity of the file]"/>
        </file-info>
<file>
```

Table 61: HTTP content server response: XML contained in the body

Please note that referring to the XML body in Table 61:

- The thumbnail part is only included if the sender uploaded a thumbnail to the server
- The validity of the files shall be specified by providing the date the files shall be removed on the server using the [ISO8601] format including the date and time in

UTC (Coordinated Universal Time) timezone (e.g. 2007-04-05T14:30:00Z). The validity depends on the configuration the originating Service Provider has set on the HTTP content server.

During the upload process the RCS client shall show the user the progress of the upload as in the case for the file transfer via MSRP.

b) If the upload is not successful, there are two cases to consider:
   i. If the server is busy and cannot handle the request a HTTPS 503 INTERNAL ERROR with *retry-after* header. The RCS client shall retry to upload after the time specified in the *retry-after* header.
   ii. If any other error, the RCS client shall automatically retry the upload as described in section 3.5.4.8.3.1.1 up to a maximum of 3 times.

5. When the upload in step 4 was successful, the sender shall then send a message to the receiver(s) with the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<file xmlns="urn:gsma:params:xml:ns:rcs:rcs:fthttp">
        <file-info type="thumbnail">
                <file-size>[thumbnail size in bytes]</file-size>
                <content-type>[MIME-type for thumbnail]</content-type>
                <data url = "[HTTP URL for the thumbnail]" until = "[validity of the thumbnail]"/>
        </file-info>
        <file-info type="file" file-disposition="[file-disposition]">
                <file-size>[file size in bytes]</file-size>
                <file-name>[original file name]</file-name>
                <content-type>[MIME-type for file]</content-type>
                <data url = "[HTTP URL for the file]" until = "[validity of the file]"/>
        </file-info>
<file>
```

Table 62: File transfer via HTTP message body content

Where compared to the body received from the content server (i.e. Table 61) an (optional) attribute has been added:

- The *file-disposition* attribute to the file-info element of the main file: This optional attribute provides functionality similar to the File-Disposition SDP attribute in file transfer via MSRP which is described in [RFC5547] and can take the same values (i.e. *render* and *attachment*). If the attribute is not included *attachment* shall be used as the default value.

Note that independently of the mechanism used to transport the message (standalone message or chat), a CPIM body will be used. As the content is now an XML, the CPIM content-type property shall be *application/vnd.gsma.rcs-ft-http+xml*.

If sending to a single user, there are two possible scenarios:

- If there is a 1-2-1 chat session established with the user and File Transfer via HTTP is supported in the session as described in section 3.5.4.8.1, the session shall be reused to convey the content shown in Table 62 in a chat message.
- There is no session established:
  - If the RCS client is configured to use standalone messaging and the recipient supports standalone messaging as well, the mentioned message body shall be delivered using a standalone message carrying a dedicated Accept-Contact header field that includes the File Transfer via HTTP IARI tag defined in section 2.6.1.1.2 along with *require* and *explicit* parameters.

- If standalone messaging is not supported by at least one of the parties, then a 1-to-1 Chat Session shall be established as specified in section 3.3.4. The RCS client shall include a dedicated Accept-Contact header field that includes the File Transfer via HTTP IARI tag defined in section 2.6.1.1.2 along with *require* and *explicit* parameters in the SIP INVITE request that it generates to establish this Chat session. The XML message shall be relayed in this session as follows:
  - If the configuration allows including the initial chat message in the SIP INVITE for a 1-2-1 chat, then it shall be used to carry the message.
  - If not, the file shall not be sent until the chat session is established.

NOTE: The inclusion of the Accept-Contact header field is only intended to guarantee that the request is routed to devices capable of File Transfer via HTTP. On the receiver's side this request can be handled as a regular invitation for Chat.

If sending to multiple users, there are two possible scenarios:

- If the file is to be transferred in an existing group chat, the session shall be reused to convey the content described in Table 52 in a chat message. If the Group Chat is closed due to inactivity, it shall be restarted first.
- There is no session established:
  - If the RCS client is configured to use standalone messaging and prior verification that all participants support standalone messaging, the mentioned XML message body shall be delivered using a standalone message with multiple recipients carrying a dedicated Accept-Contact header field that includes the File Transfer via HTTP IARI tag defined in section 2.6.1.1.2 along with *require* and *explicit* parameters.
  - If standalone messaging is not enabled a group chat session shall be established first with all the participants before sending it as a message.

When establishing a Chat Session, clients shall indicate their support for this File Transfer mechanism by including the *application/vnd.gsma.rcs-ft-http+xml* in the *accept-wrapped-types* attribute in the SDP that they provide as body in the SIP INVITE request or 200 OK response they send to take part in the Chat and include the File Transfer via HTTP IARI tag defined in section 2.6.1.1.2 in the Contact header field of that request/response. This will ensure that the conference focus does not forward the body to clients that do not support the mechanism as described in section 3.5.4.8.1.

6. The use in the client UI of the delivery notification coming from the receiver when the chat message containing the XML is delivered is left up to the RCS client implementation.

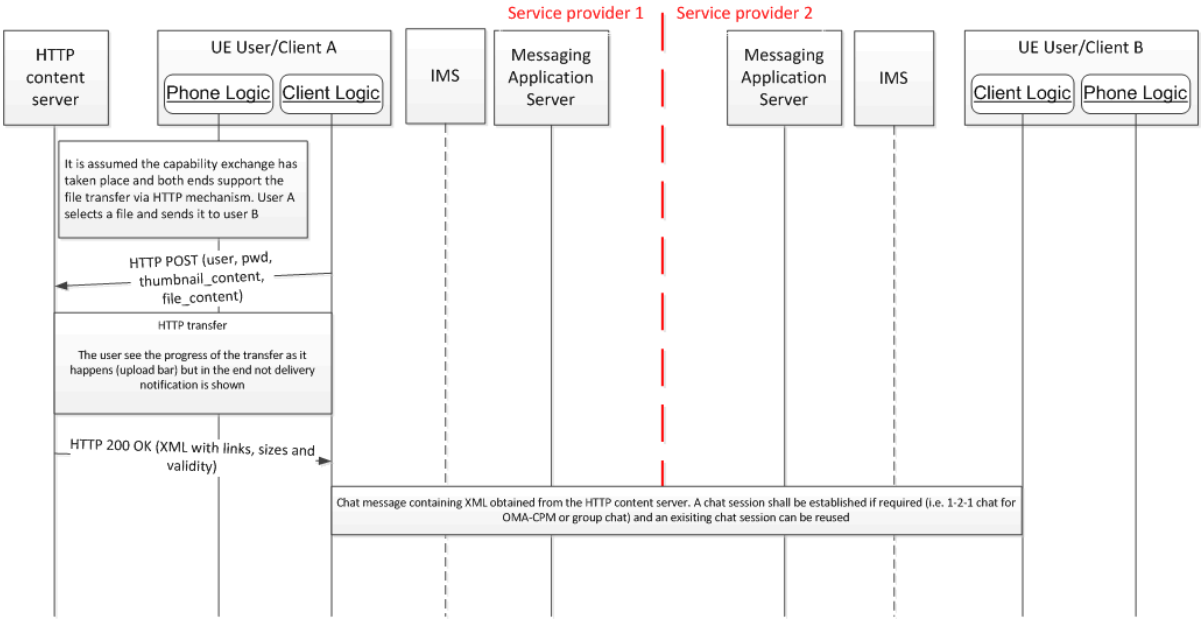


Figure 75: File transfer via HTTP: Sender procedures

Both the XML body returned by the HTTP Content Server and the optionally extended one that is exchanged between the clients shall correspond to following XML Schema which may be extended further by specific implementations and future versions of this specification. Such extensions shall be ignored by clients that are not aware of them:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:gsma:params:xml:ns:rcs:rcs:fthttp"
            xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns="urn:gsma:params:xml:ns:rcs:rcs:fthttp"
            elementFormDefault="qualified"
            attributeFormDefault="unqualified">
      <xs:element name="file">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element name="file-info" minOccurs="1" maxOccurs="2">
                              <xs:complexType>
                                    <xs:sequence>
                                          <xs:element name="file-size">
                                                <xs:simpleType>
                                                      <xs:restriction base="xs:integer"/>
                                                </xs:simpleType>
                                          </xs:element>
                                          <xs:element name="file-name" minOccurs="0"
                                          maxOccurs="1">
                                                <xs:simpleType>
                                                      <xs:restriction base="xs:string"/>
                                                </xs:simpleType>
                                          </xs:element>
                                          <xs:element name="content-type">
                                                <xs:simpleType>
                                                      <xs:restriction base="xs:string"/>
                                                </xs:simpleType>
                                          </xs:element>
                                          <xs:element name="data">
                                                <xs:complexType>
                                                      <xs:attribute name="url"
```

```
                                                type="xs:anyURI" use="required"/>
                                                <xs:attribute name="until"
                                                type="xs:dateTime" use="required"/>
                                                <xs:anyAttribute
                                                namespace="##other"
                                                processContents="lax"/>
                                        </xs:complexType>
                                </xs:element>
                                <xs:any namespace="##other" processContents="lax"
                                minOccurs="0" maxOccurs="unbounded"/>
                        </xs:sequence>
                        <xs:attribute name="type" use="required">
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:enumeration value="file"/>
                                                <xs:enumeration value="thumbnail"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:attribute>
                        <xs:attribute name="file-disposition" use="optional">
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:enumeration value="render"/>
                                                <xs:enumeration
                                                value="attachment"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:attribute>
                        <xs:anyAttribute namespace="##other" processContents="lax"/>
                </xs:complexType>
        </xs:element>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Table 63: File transfer via HTTP message body schema

3.5.4.8.3.1.1 Upload Resume

In case a file upload cannot be completed, e.g. because the file sender loses network coverage, the RCS client should allow to resume the File Transfer by using the procedure described in this section. It is intended to resume the upload of the file itself but not of an optional thumbnail which has small size. The content server shall store partial uploads and make them accessible via the related TID defined in 3.5.4.8.3.1. As it may apply a service provider policy and remove partially uploaded files after some time, resume upload may just be possible for a limited time. In case it fails, the upload cannot be resumed and the complete file needs to be uploaded again following the procedure in section 3.5.4.8.3.1. The following operations are used:

1. **Get upload info:** A client that intends to resume the upload of an interrupted File Transfer shall fetch the upload information of the file by a HTTP GET request to the content server including the TID related to the initial upload or former resume upload (see section 3.5.4.8.3.1).
   GET <FT HTTP CS URI>?tid=<tid_value>&get_upload_info HTTP/1.1
   The server sends back the upload information in the following XML structure describing

the file content without optional thumbnail including the stored byte range within a file-range tag and the direct upload URI.

```
<?xml version="1.0" encoding="UTF-8"?>
<file-resume-info xmlns="urn:gsma:params:xml:ns:rcs:rcs:fthttpresume">
        <file-range start="[start-offset in bytes]" end="[end-offset in bytes]" / >
        <data url="[HTTP upload URL for the file]"/>
</file-resume-info>
```

Table 64: File transfer via HTTP upload information content

Complying with following schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:gsma:params:xml:ns:rcs:rcs:fthttpresume"
            xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns="urn:gsma:params:xml:ns:rcs:rcs:fthttpresume"
            elementFormDefault="qualified"
            attributeFormDefault="unqualified">
        <xs:element name="file-resume-info">
            <xs:complexType>
                <xs:sequence>
                        <xs:element name="file-range">
                                <xs:complexType>
                                        <xs:attribute name="start" type="xs:integer"
                                        use="required" />
                                        <xs:attribute name="end" type="xs:integer"
                                        use="required" />
                                        <xs:anyAttribute namespace="##other"
                                        processContents="lax"/>
                                </xs:complexType>
                        </xs:element>
                        <xs:element name="data">
                                <xs:complexType>
                                        <xs:attribute name="url" type="xs:anyURI"
                                        use="required"/>
                                        <xs:anyAttribute namespace="##other"
                                        processContents="lax"/>
                                </xs:complexType>
                        </xs:element>
                        <xs:any namespace="##other" processContents="lax" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
</xs:schema>
```

Table 65: File transfer via HTTP upload information schema

In case of a successful HTTP response by the server, e.g. HTTP 200, including an XML description of the file, the following procedure applies depending on the content of the XML description:

- If it includes file-resume-info for the uploaded file content with file range which matches the original file size, the file has been uploaded successfully.
- If it includes file-resume-info of the uploaded file content but with file range below the file size, the remaining file content needs to be uploaded using step 2.
- If it does not include the file-resume-info of the file content, the full upload needs to be started from beginning using the HTTP POST request as described section 3.5.4.8.3.1.

NOTE: The file-range refers to the part of the file that has been uploaded prior to the resume upload.

A server shall send back an HTTP error response if resume upload cannot be performed (e.g. because the partial files are no longer available) according to [RFC2616], e.g. HTTP 404 or 410. An HTTP response that does not contain an XML description of the file or an XML structure that does not include a range field, shall indicate to the client that a resume of the upload of the file is not possible and that therefore a full upload needs to be done again.

2. **Resume upload**: In case the client wants to resume the upload of the file content it generates an HTTP PUT request to the upload URL that was included in the XML description provided by the content server in operation 1. In this request it shall provide the remaining bytes started from the already uploaded byte position that was included in the received XML description. To indicate the byte range that is included in the HTTP PUT request a HTTP *Content-Range* header as defined in [RFC2616] is added to the request:

```
PUT http://<file_upload_uri> HTTP/1.1
Content-Type: [mime type depending on the file; e.g. image/jpeg]
Content-Length: <remaining_upload_size>
Content-Range: bytes <first-byte-pos> - <last-byte-pos> / <file_size>
Authorization: Digest ...

<file content>
```

Table 66: File transfer via HTTP upload information content

When the server receives the partial file, it shall append the data according to the Content-Range header. In case the upload is successful, a HTTP 200 OK response without body is returned.

The client has to ensure that the file content related to the TID has not been changed between the initial HTTP POST request and the resume upload operation.

NOTE: Also this HTTP PUT can fail, e.g. due to another loss of network coverage. In that case the operations 1 and 2 may be repeated with the same TID. In that case the file-range tag indicates the sum of all the data uploaded in the uploaded resumes that have taken place so far.

3. **Get download info**: To get the XML description of the complete file to be sent to the file receiver according to 3.5.4.8.3.1, the client sends the following request to the content server:
GET <FT HTTP CS URI>?tid=<tid_value>&get_download_info HTTP/1.1
The server sends back a successful HTTP response including the XML description back if the file has been uploaded successfully. In that case the XML includes the file info for the thumbnail (if provided) and the file (as defined in Table 61). Otherwise an HTTP error response will be returned.

NOTE: Like for the initial HTTP POST in section 3.5.4.8.3.1 authentication may be requested for other HTTP requests used in this section. In that case the client shall sent it a second time carrying the authentication header field in line with the challenge received in the HTTP 401 AUTHENTICATION REQUIRED response to the first request. All additional HTTP requests towards the content server shall use the HTTP digest authentication as

defined for regular file upload. In case HTTP over TLS is used, HTTP basic authentication can be used instead.

The whole procedure (including the initial upload is summarized in following figures:

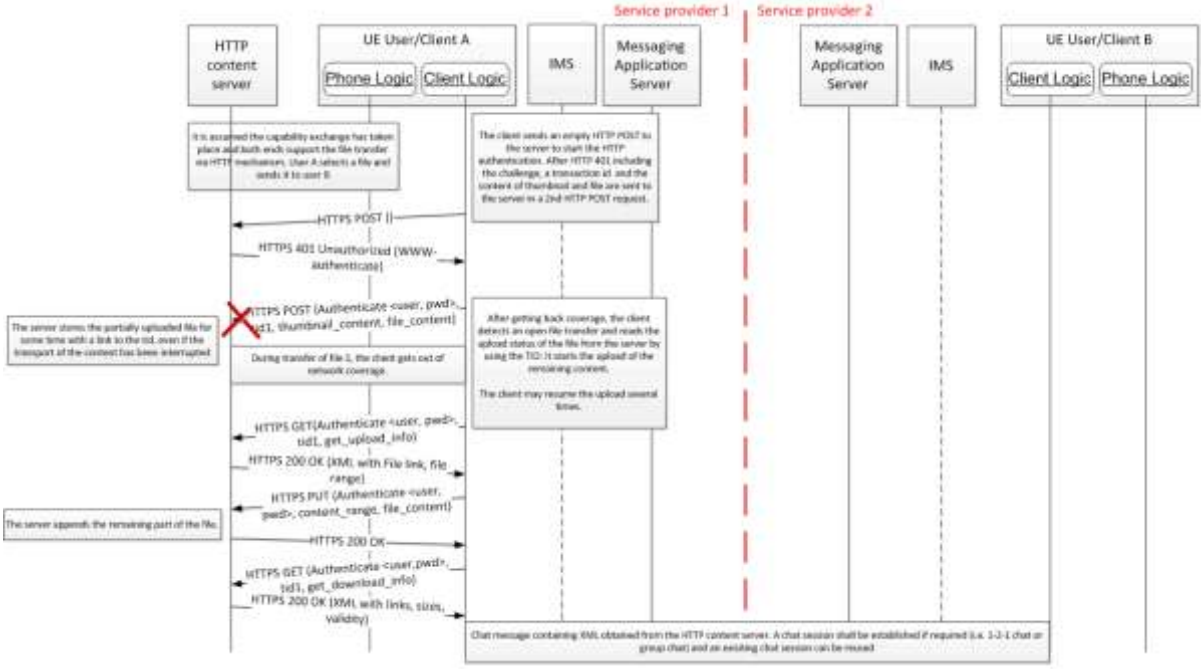


Figure 76: File transfer via HTTP: Resume upload

In case the resume is not possible (anymore), the flow shall be as follows:

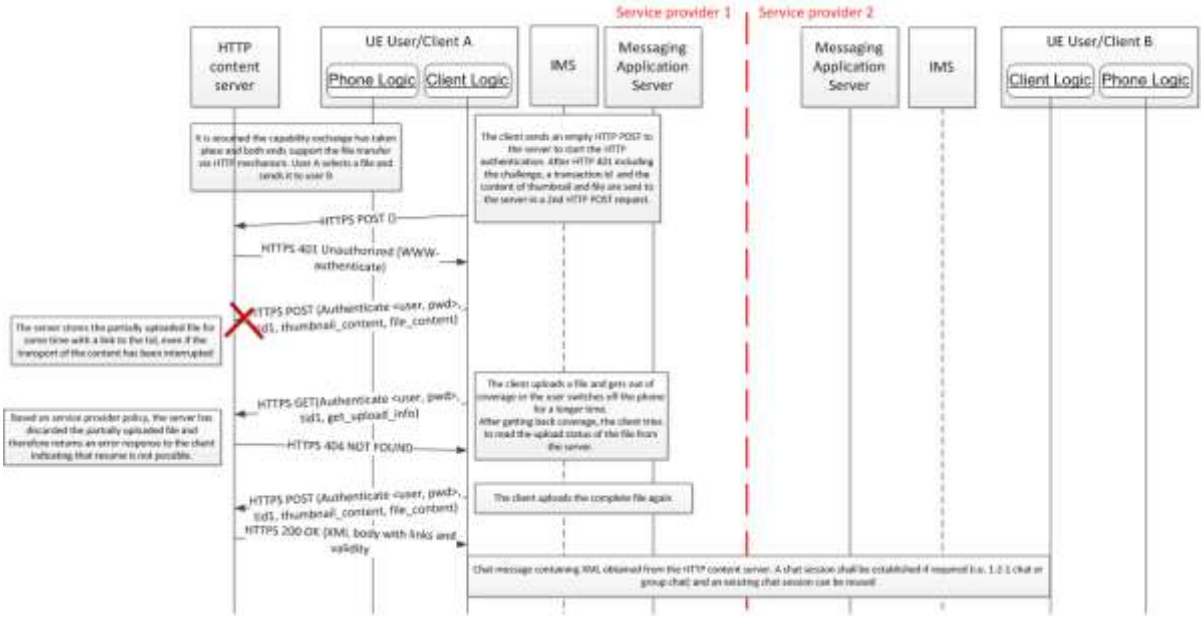


Figure 77: File transfer via HTTP: Resume upload not possible

3.5.4.8.3.2 Receiver procedures

When the receiver gets a chat message as described in the previous section, the RCS client shall:

1. The user shall not be aware a different procedure has been used to carry the file, therefore and, if present, the RCS client shall download (HTTPS GET) the thumbnail and display/notify of the incoming file transfer.
2. If the user accepts, the file shall be downloaded (HTTPS GET) showing the progress of the download as for a file transfer performed for MSRP. If the HTTP content server is working adequately, one of the following three responses shall be returned to the client:
   - HTTP 200 OK: Meaning the file is downloaded The client shall handle the file then according to the file-disposition attribute if included in the File transfer via HTTP message body content (see Table 62 and section 3.5.4.8.3.1).
   - A HTTP 503 INTERNAL SERVER ERROR with a Retry-After header: In this case the client shall retry, the recommended value to retry will be specified in the “Retry-After” header. Please note that this response is provided by the server when the sender is still uploading the file to prevent the race condition
   - Any other error: The client shall retry up to a maximum of 3 times. In case the file was partially downloaded already, a partial HTTP GET request as defined in [RFC2616] may be used to obtain the remaining part of the file.
3. Regarding the display notification associated to this chat message, it shall only be sent when the file has been successfully downloaded to indicate the sender that the file has been effectively downloaded by the user.

Finally note that if validity of the file to be downloaded indicates that it may no longer be available on the server, the client shall inform the user of the circumstance when trying to download the file. The detailed UX is left intentionally outside the scope of this specification and it is up to the RCS client implementation.

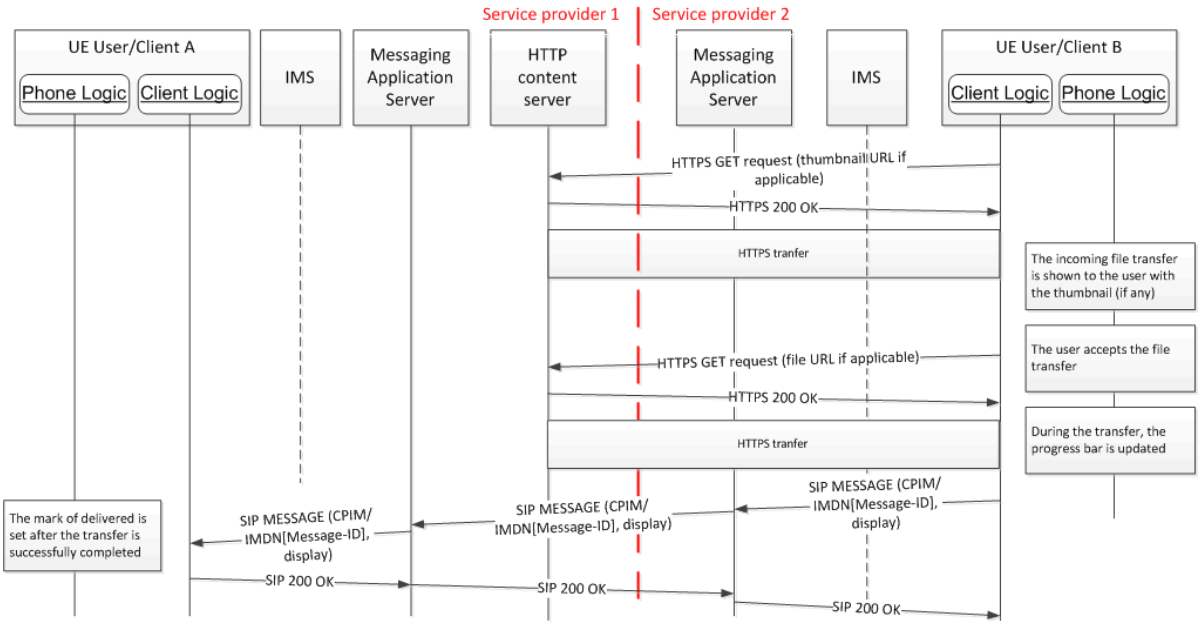


Figure 78: File transfer via HTTP: Receiver procedures

3.5.4.8.3.2.1 File transfer auto-accept

Consistently with Annex A sections A.1.4 and A.2.6, if the parameter *FT AUT ACCEPT* is set to 1 and the indicate file size is smaller than the size configured in the *FT WARN SIZE* configuration parameter, the receiving client shall not only download automatically the thumbnail but also the file content.

3.5.4.8.4 HTTP Content server addressing

In order to enable the traceability of the HTTP transactions among operators, the HTTP content server FQDN shall follow the format presented below:

ftcontentserver.rcs.mnc<MNC>.mcc<MNC>.pub.3gppnetwork.org

Table 67: HTTP content server FQDN

3.5.4.8.5 Security considerations

In order to guarantee the integrity and security of the solution for file transfer via HTTP the following three principles shall be taking into account:

1. The security of the solution relies on the security of the chat messages. Therefore, encryption of the media associated to Chat (1-to-1/Group Chat) media is recommended.
2. All HTTP transactions shall be secured using HTTPS.
3. To secure interoperability between Service Providers and to reduce complexity on the RCS device/client, the HTTP configuration server shall make use of public root certificates issued by a recognized CA. That is the root certificates are similar to those used by standard webservers which are widely recognized by browsers and web-runtime implementations both in PCs and devices.

3.5.4.8.6 A Common File Store

3.5.4.8.6.1 Overview

A Common File Store for the files may be used side by side with a Common Message Store for the chat messages carrying the XML with the information for the thumbnail (optional) and the file. As for the Common Message Store, the Common File Store may be used to synchronize files between devices. It also allows the user to keep a back-up of files in the network. Note, that in this deployment scenario the Common File Store stores the media objects whereas the Common Message Store stores the chat messages that contain links to that media.

The Common File Store is meant both for sent and received files and requires the existence of a Common Message Store. For the Common File Store to work with the Common Message Store a function named File Transfer Localization function is required.

The File Transfer Localization function shall always be triggered upon a client/device request to download a file. This is achieved through the Messaging server overwriting the HTTP URL for the thumbnail (if provided) and the HTTP URL for the file with an HTTP URL pointing to the File Transfer Localization Function. As for the values of the other XML parameters (e.g. until = "[validity of the file]”), they are unchanged and remain relevant.

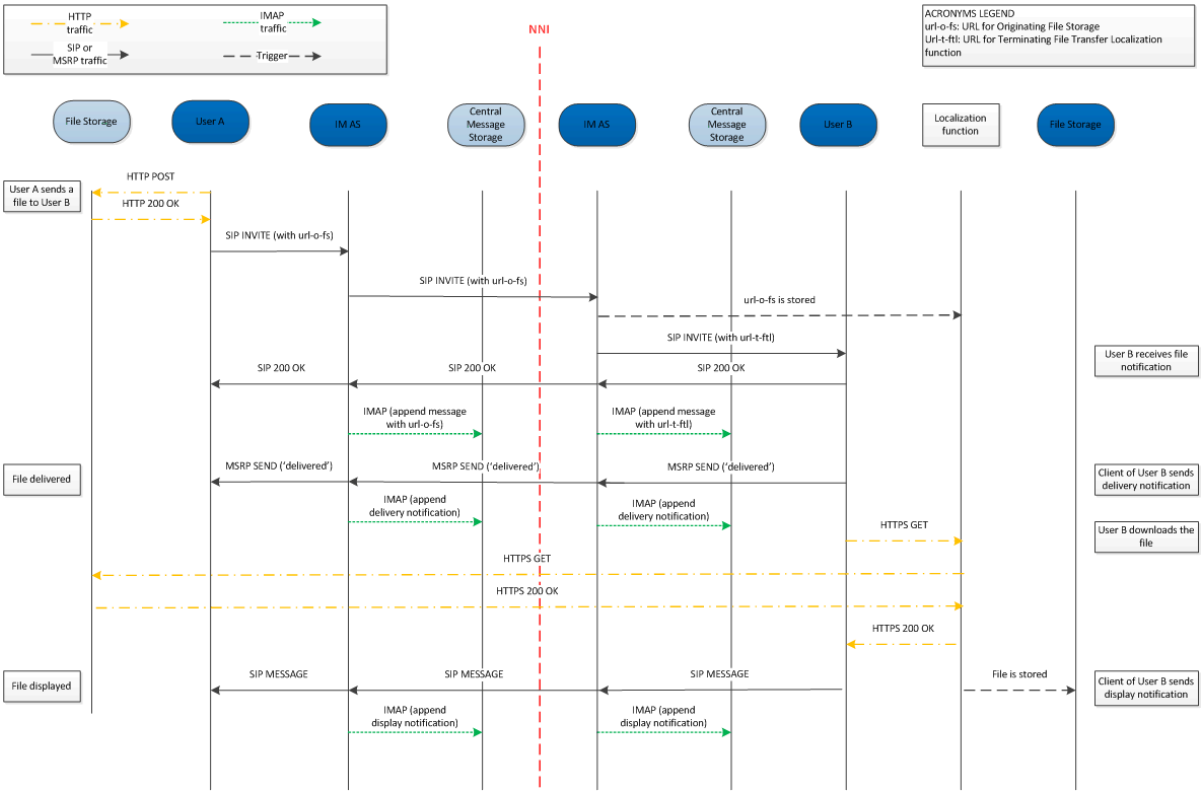


Figure 79: File Transfer Localization Function: global flow

3.5.4.8.6.2 Sender Procedures

The Service Provider shall direct the files uploaded by the sender towards a permanent storage with or without triggering the File Transfer Localization function. The Same or a different location can be used for uploading files sent by legacy clients.

The procedures as described in 3.5.4.8.3.1 apply. Please note that referring to the XML body in Table 61 the following should be taken into account:

- The validity indicates how long the file is available to the receiver for first time file download request. For the messages retrieved from the Message Store Server the validity is not relevant and should be ignored because permanent storage applies.

The upload resume procedures described in 3.5.4.8.3.1.1 remain relevant.

For other devices of the user, the retrieval procedures defined in 3.5.4.8.6.4 apply.

3.5.4.8.6.3 Terminating Messaging Server Procedures

As described in 3.5.4.8.6.1, a receiver's request to download a file shall always trigger the File Transfer Localization function. The HTTP URL for the thumbnail (optional) and the HTTP URL for the file included in the XML received by the client shall direct the HTTP GET request towards the File Transfer Localization function.

The Messaging Server shall overwrite any URL contained in the *url* attribute of the data element of a File transfer via HTTP message body content for both the thumbnail (if included) and the file itself.

3.5.4.8.6.4 Receiver Procedures

The receiver procedures are unchanged compared to section 3.5.4.8.3.2 when receiving the File Transfer request from another user.

Note that the validity shall indicate to the client how long the file is available for first time download request. Once the client sends the file download request, the file is localized and made available for future retrievals. For the messages retrieved from the Message Store Server validity is not relevant and should be ignored as permanent storage applies.

Once the user sends the HTTP GET request for first download or subsequent file retrieval, a similar authentication procedure as described in section 3.5.4.8.3.1 may be initiated. Specifically, the HTTP GET request towards the Localization Function can result in a HTTP 401 AUTHENTICATION REQUIRED error response carrying a WWW-Authenticate header field as defined in [RFC2616]. The receiver shall in this case depending on the WWW-Authenticate header received then make another HTTP GET request using basic authentication or HTTP digest as per [RFC2617] using *FT HTTP CS USER* and *FT HTTP CS PWD* configuration parameters as credentials (defined in Table 86 in section A.1.4).

NOTE: It is left to Service Provider to ensure that authentication procedure is not triggered towards legacy clients.

3.5.4.8.6.5 File Transfer Localization function procedures

The File Transfer Localization function, once triggered, shall either download the file from the location that the file sender has uploaded it (first time download) or retrieve it from the Common File Store (any subsequent download). For the former case, the File Transfer Localization function shall also upload the file to the Common File Store so as to be available for any subsequent download requests coming from a different device of the same user.

The File Transfer Localization function shall keep track for every file transfer of the HTTP URL provided by the sender so as to be able to download the file upon first client download request. Whether or not the File Transfer Localization function is triggered for files sent to a legacy device is left up to the Service Provider implementation.

3.5.4.9 Handling of specific content

3.5.4.9.1 Personal Card format

Current implementations of the vCard standard by different device manufacturers leads today to data loss of certain contact information, when this information is exchanged among devices or synced with network address books. An RCS compliant device shall support receiving at a minimum, vCard 2.1 [vCard21] and vCard 3.0 formats [RFC2425], [RFC2426] and may support also the Personal Contact Card (PCC) format [CAB_TS].

The following fields are considered key fields. No data of these fields should be lost when contact information is exchanged by any means (peer to peer contact sent, uploaded, synchronized, etc.):

- Name
- Telephone numbers
- Email addresses
- Address information
- Personal information

The Minimum subtypes that should be supported are defined in the PCC definition in [CAB_TS]:

- Name: Composed names (such as “Jean-Baptiste”) shall be supported properly
- Personal Information
  - Nickname
  - Photo

  - Birthdate
  - Comment
- Telephone number: At least the following subtypes of telephone number shall be supported:
  - Land home
  - Land work
  - Land other
  - Mobile home
  - Mobile work
  - Mobile other
  - Fax work
  - Fax other
  - Beeper
  - Other
- Email addresses: The following subtypes shall be supported:
  - Email work 1
  - Email work 2
  - Email home 1
  - Email home 2
  - Other
- Address information
  - Address
  - Geographic Position
  - Timezone

Sending and receiving a contact card via File Transfer is technically the same as sending any other file.

If the format for pushing a contact card file is vCard 2.1 or 3.0 formats, the MIME (Multipurpose Internet Mail Extensions) type that shall be used for the file transfer is "*text/vcard*".

If the format for pushing the contact card is CAB (Converged Address Book) 1.0 PCC XML format, then the CAB PCC MIME type "*application/vnd.oma.cab-pcc+xml*" shall be used.

On the receiving side, after the receiving RCS user accepts the contact card file delivered through File Transfer, the receiving RCS client shall apply the mapping of the RCS supported fields between the received format (CAB PCC XML for example) and the used format of the local address book database[29].

vCard 3.0 format is recommended in RCS.

[29] If the conversion between PCC and vCard is required, please see [CAB_TS] section 5.4.3 "Format Adaptation".

If the receiving side does not support the offered format identified in *the a=file-selector* attribute of the SIP INVITE SDP, it should reject the File Transfer invitation with an error response indicating it does not support the content-type, which then causes the sending side to initiate a second File Transfer, this time sending the contact card in a different format.

3.5.4.9.2 Audio Message

The handling of audio messages is described in section 3.11.4.

3.5.5 NNI and IOT considerations

In addition to what is defined in Section 2.12, the mapping of the appropriate File Transfer feature tags is done by the Messaging Server, as per Appendix G in [RCS5-CPM-CONVFUNC-ENDORS] when it is determined that the remote network requires such interworking.

3.5.6 Implementation guidelines and examples

From the UX perspective there are five possible entry points to this service:

1. Address book/Call-log: A file transfer can be initiated with any registered contact providing the correct capabilities are in place. This is contact oriented initiation. Following the address book interaction, the list of available files is displayed allowing the user to select one or more files to share. Once the file transfer commences, the progress can be checked in the standard notification area.

Figure 80: Reference UX for accessing file share from address book/call-log

2. Media gallery/File browser: The user can browse, select a file (or multiple files) and then share these with one or more RCS users. This is task contact oriented initiation. Only RCS capable users shall be displayed as candidate recipients of the file.

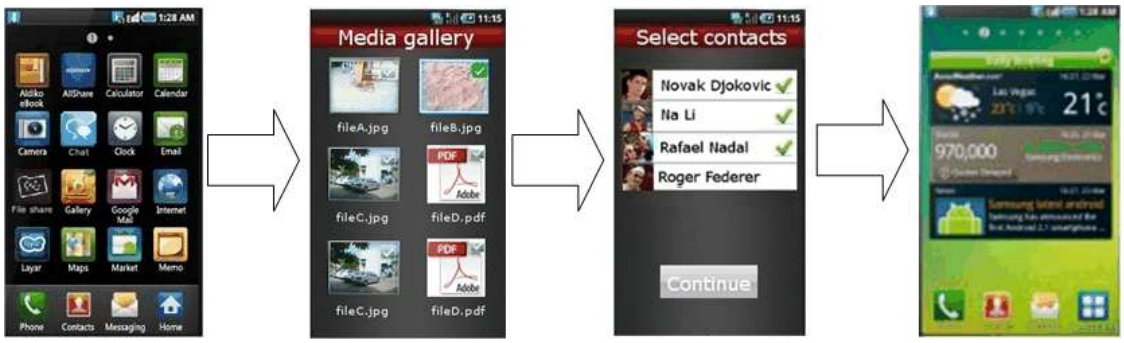


Figure 81: Reference UX for accessing file share from media gallery or file browser

In the previous figure, once File Transfer is selected, the user will be presented with the complete list of RCS contacts (including contacts which are currently not registered).
In this case, a SIP OPTIONS or Presence exchange is triggered once a contact is selected from the list.

3. Camera application: The experience is similar to the media gallery/file browser experience with the difference being that the user is able to select only the last picture or video (and, in some cases, a picture or video from the camera gallery) to be shared.
4. Chat window: From the Chat window a file can be shared using the relevant button/icon. The experience is identical to the address book/call-log. The user is redirected to the media gallery or file explorer where the user can choose a file which, is then shared with the conversation partner(s).

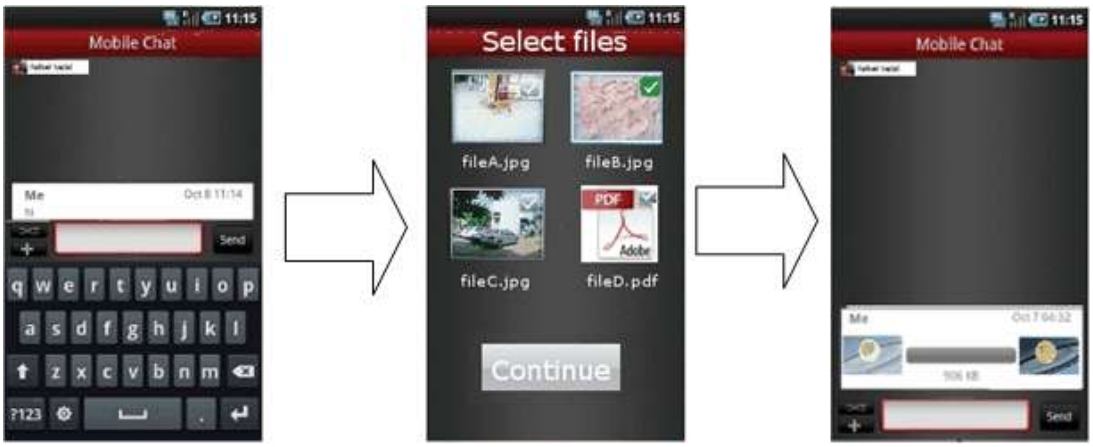


Figure 82: Reference UX for accessing file share from a Chat window

5. Call screen (Image Share): a picture can be shared either from the camera (front or back) or by choosing a file from the media gallery. Please note this case has been covered in detail in section 3.6.6.1.2.

3.5.6.1 Handling of specific content

3.5.6.1.1 Personal Card handling

The personal and business cards of the RCS user may be stored in a way that is compliant to the CAB 1.0 PCC data in the RCS client which enables the RCS user to create and populate any number views on the personal and/or business contact information as needed. A client may tag these with their dedicated purposes (professional, friends, etc.).

A Personal Card is, from a technical perspective, the same as any other contact card. This functionality only requires certain user experience changes. In particular:

- Visibility as an option in the address book menu.
- A special name/mark in the address book to easily distinguish it from the rest of the contacts.

It is recommended to support at least three Personal Cards. In particular:

- Business Card: For professional use.
- Two more Personal Cards to allow social uses (e.g., a contact card to be exchanged with closest friends for having fun, including frequently updated fields such as a personal picture) and an additional one to allow having a stable personal profile for non-professional uses.

3.5.6.1.2 Personal Contact Card entry points

Sending a contact card

The user selects any of the contacts in the address book. Entry points for sending a contact could be:

1. Chat
2. address book
3. call log

Before sending a contact card the user should have the option to preview the information. The possibility of editing the information should be available so that filtering the contact information to be sent is also allowed. Once the contact information is confirmed the contact card is sent over File Transfer.

Receiving a contact card

When a new contact card is received, the user is prompted to accept the file. Once accepted, two options are given:

1. Save contact card
2. View contact card

If 'Save Contact information' is chosen proper options will be given depending on whether the contact received already exists or not in the receiver's address book. If it exists the existing contact information will be implemented with the additional information received.

3.5.6.1.3 Audio Messages

The entry points for audio messages are described in section 3.11.6.

3.6 Content sharing

3.6.1 Feature description

3.6.1.1 Overview

Content sharing provides the capability to share videos and pictures in near real-time. This functionality can be used both in connection to a voice call and in a standalone manner when there is not an ongoing call. When the receiving user has multiple devices the content sharing requests are sent to all those devices. Therefore when used in combination with a voice call, the user can decide to accept the shared content on a different device than the one they are using for the voice call if that device has better display capabilities for instance.

There can be different sources for the shared content:

- The front camera ("me")
- The rear camera ("what I see")
- A file ("video streaming" or "sending of a stored image")

A Service Provider configurable parameter allows the Service Provider to set the maximum duration of a Video Share session (see VS MAX DURATION in section A.1.5) and the max size of a file transferred during Image Share (see IS MAX SIZE in section A.1.5).

From the user experience perspective and assuming that the duration and size limitations are in place (i.e. the values are non-zero):

- When performing a video share, if the session duration (send or receive) is approaching the VS MAX DURATION, a warning notification could be displayed for the user.
- When performing a video share, if the session duration (send or receive) is longer than the VS MAX DURATION, a warning message will be displayed and the video share

session will be cancelled (that is at protocol level, the SIP BYE request will be sent to the other end).

- If the picture size in an image share session is bigger than IS MAX SIZE, a warning message will be displayed when trying to send or receive a picture larger than the mentioned limit and the image share session will be cancelled (that is at protocol level, the SIP INVITE request will never be sent or an automatic rejection response will be sent to the other end depending on the scenario).

3.6.1.2 Content Sharing during a voice call

The content share services during a voice call are linked to the call. Therefore they are also automatically terminated when the call ends.

All services are delivered as one to one only and there is no multiparty sharing provided. For the content sharing during a call, the user should be able to recognize whether one or both content sharing services (i.e. Video and Image Share) are available to use with their conversation partner. Therefore both ends need to be updated on the respective capabilities to avoid showing a service as available when this is no longer the case. This is achieved through the capability exchange described in section 2.6.

Both Video and Image Share are unidirectional and do not need a dedicated audio path. It is possible however to establish simultaneous Image and/or Video Share sessions in each direction. For example when referring to bidirectional Video Share, this means that once User A is sharing video with User B, User B can also start to share video with User A simultaneously, provided the right coverage conditions are in place. In this case each Video Share session is independent and should be handled separately. When a user's coverage conditions change while such bidirectional sharing is active, the device that changed coverage shall terminate the sharing session that it initiated. The same example would also apply to Image Share or to a combination of Video Share in one direction and Image Share in the other.

3.6.1.3 Content Sharing without a voice call

For the sharing without a call the user is aware which services are available through the regular RCS capability query mechanisms defined in section 2.6. For sharing files or images the File Transfer service as described in section 3.5 is used.

NOTE: It is possible to use the content sharing without a voice call also with the conversation partner during a voice call. In that case, the sharing session will be independent of that voice call. This means that it can continue after the voice call ends and must be explicitly terminated. Furthermore for live video sharing, audio will be sent in the voice call in addition to the audio stream that is part of the sharing session.

Similar to the content sharing during a call, the Video Share session is unidirectional. However it is not possible to establish simultaneous Video Share sessions with the same or different users as that might result in user experience issues such as the audio from one session being retransmitted in another. To establish a Video Share session in the other direction, the already established session must therefore be terminated first.

Users are allowed to qualify undesired incoming Video Share requests without a voice call as spam. To this end, clients may support a locally stored blacklist to handle incoming Video Share requests without a voice call. This is the same blacklist as it is used for incoming chat and file transfer requests. If an invitation to accept a live video without a voice call is received from a blacklisted user, the client should reject the Video Share request. Instead it may log the event in the spam folder(e.g. "User A tried to share a video on TIME/DATE").

3.6.1.4 Use Cases

3.6.1.4.1 Share Video during a voice call

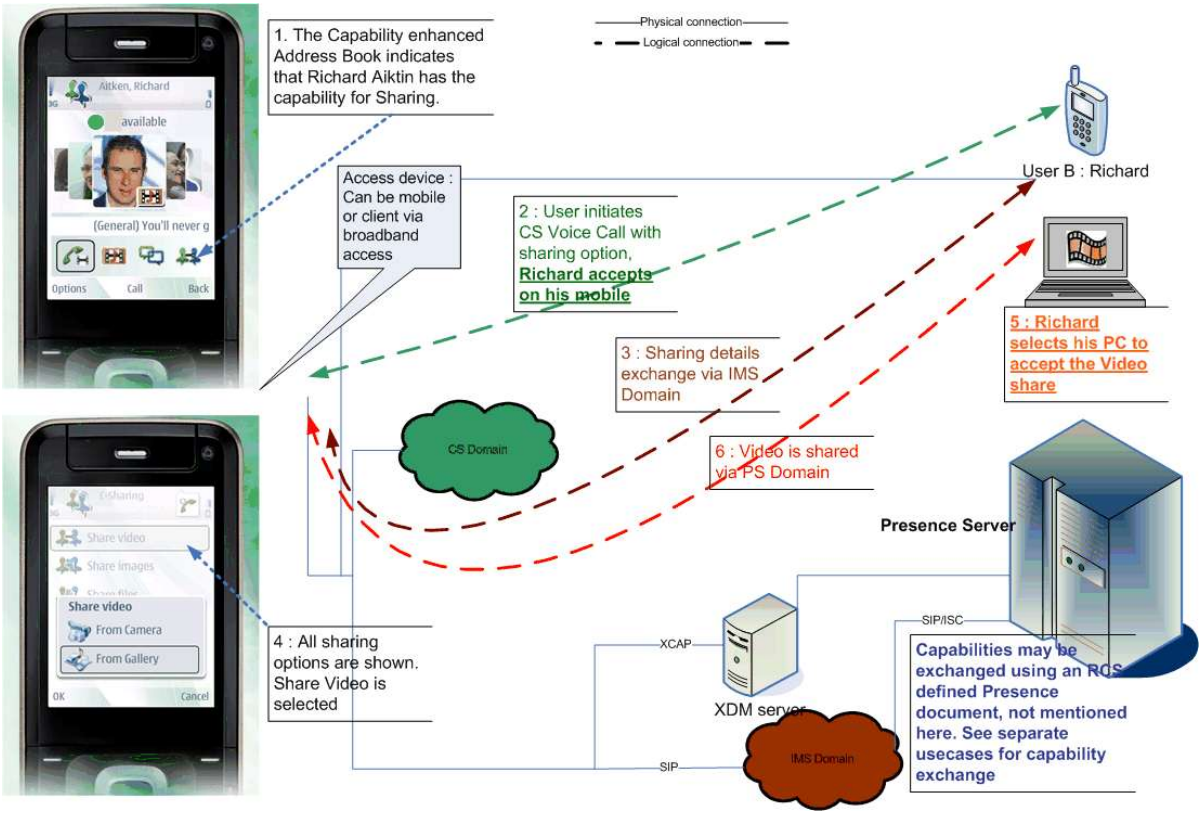


Figure 83: Sharing video during a voice call

Figure 83 illustrates the behaviour when the voice call is set up in the CS domain. Apart from the voice call itself, the behaviour would be identical though if one or both parties used the PS domain for the voice call as specified in section 3.8 since the sharing service functions independently of the voice call.

NOTE: When both of the devices involved in the sharing are on a high bandwidth access, for example LTE, the perceived video quality will be higher.

3.6.1.4.2 Sharing video during a call in the multidevice environment

User A has a mobile device and a broadband access device (RCS PC client). User B has a mobile device.

- User B has travelled to Hong Kong and is visiting the Victoria's peak. The view from top of the peak is astonishing and they would like to share the experience with their friend User A.
- User B makes a call to User A
- User A answers on the mobile.
- User B tells User A about the view they are viewing. To prove this User B decides to share a video with User A.
- User B sees from the call menu that they can share video with User A. User B sends the request to share video, for example, by clicking the Video Share icon.
- The request is sent to both User A's mobile and PC; both mobile and PC will alert.
- As User A is sitting in front of their PC he/she decides to take the video to the PC for example, by clicking accept button on the PC client.

- User A's mobile will then stop alerting.
- User A will now see the beautiful scenery shared by User B in their PC while still having the voice call on the mobile.

  NOTE: this was illustrated previously in Figure 83. The behaviour would be similar when sharing an image.

3.6.1.4.3 Share an Image during a call

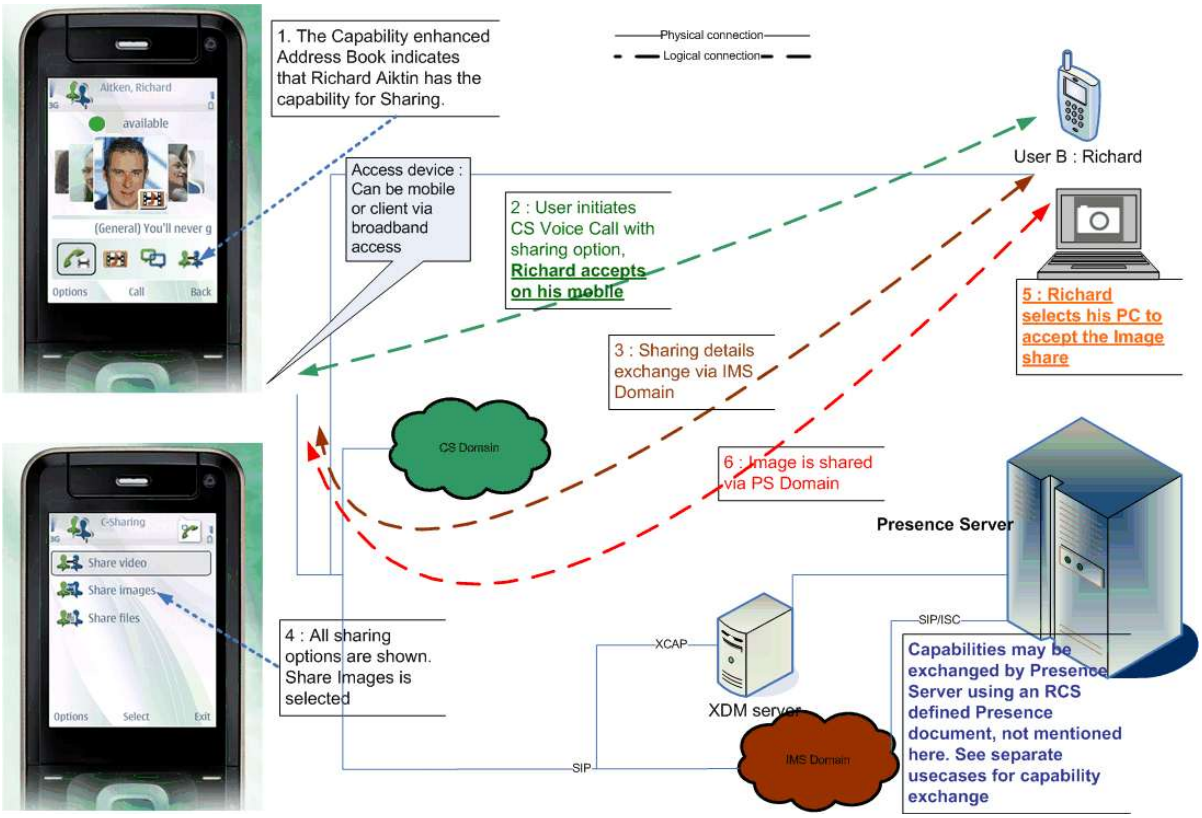


Figure 84: Sharing an image during a call

Figure 84 illustrates the behaviour when the voice call is set up in the CS domain. Apart from the voice call itself, the behaviour would be identical though if one or both parties used the PS domain for the voice call as specified in section 3.8 since the sharing service functions independently of the voice call.

3.6.1.4.4 Share a video without a voice call

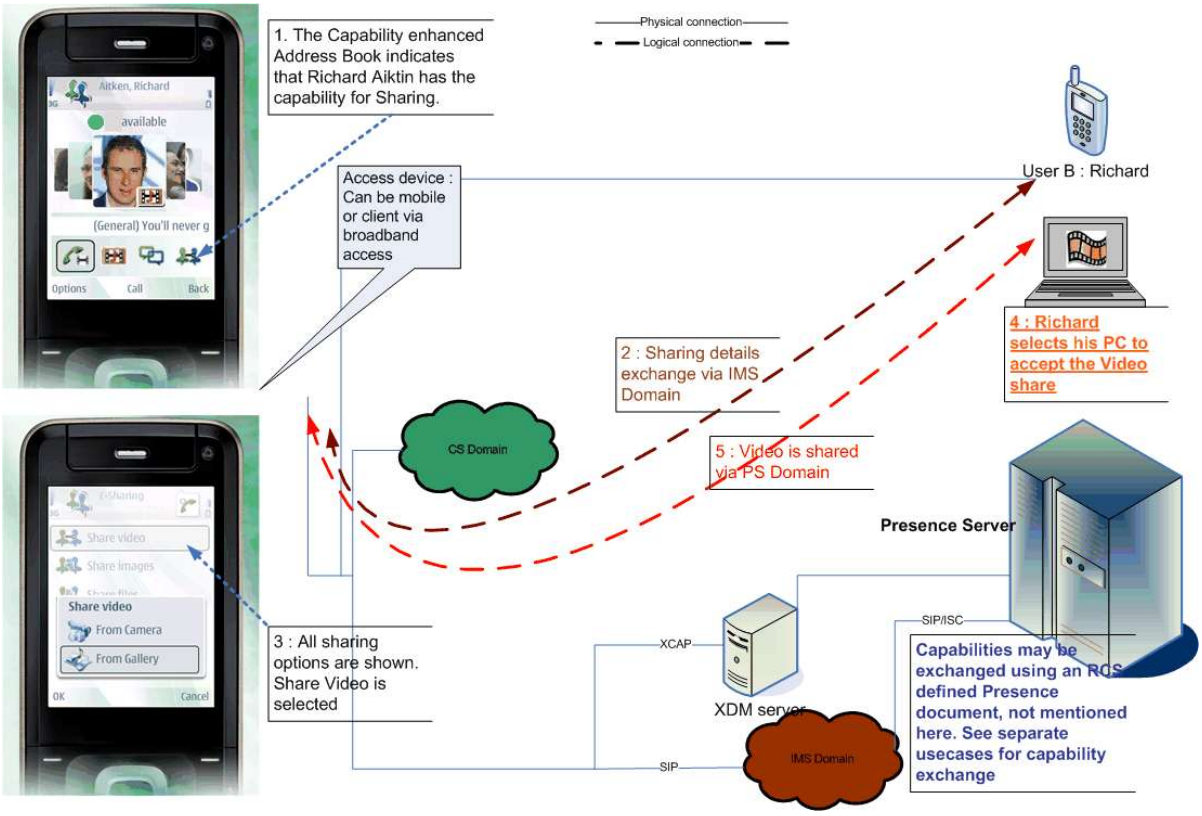


Figure 85: Sharing video without a call

NOTE: When both of the devices involved in the sharing are on a high bandwidth access, for example LTE, the perceived video quality will be higher.

3.6.2 Interaction with other RCS features

3.6.2.1 Voice Call

The sharing during a voice call (either over CS or as specified in section 3.8) interacts with that voice call since the sharing is automatically terminated when the call is terminated. There is also an interaction with the supplementary services of that voice call.

3.6.2.1.1 Multiparty call and Image/Video Share

Once a voice call is established between two users, it is possible for one of them to add another party to the call, and consequently, initiate a multiparty call. From RCS services perspective and as presented in section 2.6.4.1, the Image and Video Share services are not available during a multiparty call. Therefore the terminal should manage the following scenarios:

- The users were in a voice call without using the Image or Video Share services: In this case, when switching to a multiparty call the client starting the process has to send a SIP OPTIONS request with a capability update (as described in section 3.6.4.3.2) indicating that the Content Sharing services during a call are no longer available. The on-screen icons/layout should be updated accordingly.
- The users were in a voice call using Video Share: In this case, switching to a multiparty call means ending the Video Share service. This can either be sender or receiver terminated, depending upon the circumstances, as described in sections 3.6.4.3.4 and 3.6.4.3.5 respectively. In both cases, a capabilities exchange using SIP occurs and, consequently, the client initiating the multiparty call should report that the Content

Sharing services/capabilities during a call are no longer available. The on-screen icons/layout should be updated accordingly.

- The users were in a voice call using Image Share with the transfer not yet completed: In this case, switching to a multiparty call means ending the Image Share service. This either can be sender or receiver terminated, depending upon the circumstances, as described in sections 3.6.4.3.8 and 3.6.4.3.9 respectively. In both cases, a capabilities exchange using SIP OPTIONS occurs and, consequently, the client initiating the multiparty call should report that the Content Sharing services/capabilities during a call are no longer available. The on-screen icons/layout should be updated accordingly.
- The users were in a voice call using Image Share after the transfer has completed: In this case, switching to a multiparty call means that the picture is no longer shown in the call screen and that the client starting the process has to send a SIP OPTIONS message with a capability update (as described in section 3.6.4.3.2) indicating that the Content Sharing services during a call are no longer available. The on-screen icons/layout should be updated accordingly.

It should be also noted that from the moment the users enter in a multiparty call, it is not necessary to perform the capability exchange described in section 3.6.4.3.2.

Finally, if the multiparty call is converted into a standard call (That is it becomes again a 1-to-1 call), this event should be treated as a new call establishment meaning that a capability exchange via OPTIONS needs to take place and, consequently, the relevant on screen icons need to be updated.

3.6.2.1.2 Call on hold and Image/Video Share

Once a voice call is established between two users, it is possible for one of them to put the other party on hold. From RCS services perspective and as presented in section 2.6.4.1, the Image and Video Share services are not available during a call which is not active, therefore, the terminal needs to manage the following scenarios:

- The users were on a voice call without using the Image or Video Share services: In this case, when putting the call on hold the client starting the process has to send an SIP OPTIONS request with a capability update (as described in section 3.6.4.3.2) indicating that the Content Sharing services during a call are no longer available. The on-screen icons/layout should be updated accordingly.
- The users were in a voice call using Video Share: In this case, putting the call on hold means ending the Video Share service. This can either be sender or receiver terminated, depending upon the circumstances, as described in sections 3.6.4.3.4 and 3.6.4.3.5 respectively. In both cases, a capabilities exchange using SIP OPTIONS occurs and, consequently, the client putting the call on hold should report that the Content Sharing services/capabilities during a call are no longer available. The on-screen icons/layout should be updated accordingly.
- The users were in a voice call using Image Share with the transfer not having completed: In this case, putting the call on hold means ending the Image Share service. This can either be sender or receiver terminated, depending upon the circumstances, as described in sections 3.6.4.3.8 and 3.6.4.3.9 respectively. In both cases, a capabilities exchange using SIP OPTIONS occurs and, consequently, the client putting the call on hold should report that the Content Sharing services/capabilities during a call are no longer available. The on-screen icons/layout should be updated accordingly.
- The users were on a voice call using Image Share after the transfer has completed: In this case, putting the call on hold means that the picture is no longer shown in the call screen and that the client starting the process has to send a SIP OPTIONS message with a capability update (as described in section 3.6.4.3.2) indicating that the Content

Sharing services during a call are no longer available. The on-screen icons/layout should be updated accordingly.

It should be also noted that from the moment the call is put on hold (that is the call is not active):

- It is not necessary to perform the capability exchange described in section 3.6.4.3.2, and,
- If there is another active call, the behaviour regarding the Image and Video Share services (that is both for the capability exchange and the services itself) should not be affected by the fact that another call is on hold.

Finally, if the call is made active, this event should be treated as a new call establishment meaning that a capability exchange via OPTIONS needs to occur and, consequently, the relevant on screen icons need to be updated.

3.6.2.1.3 Waiting call and Image/Video Share

A waiting call is a non-active call; therefore, consequently with the information presented in section 2.6.4.1, it should not be possible to access the Image and Video Share services between the caller and receiver.

Please note having a waiting call will not affect the behaviour for Image and Video Share (that is both for the capability exchange and the services itself) on the active call.

3.6.2.1.4 Calls from private numbers

When a call is received and the caller cannot be identified (because a hidden number is used for instance), it should not be possible to access the Image and Video Share services between the caller and receiver.

3.6.2.1.5 Call divert/forwarding

If the receiver has call divert/forwarding active (the calls are for instance forwarded to another number or to voicemail), it is not possible to access the Image and Video Share services from the caller to the receiver.

3.6.2.2 Chat

As for the sharing without a call there is not necessarily a communication context allowing the receiving user to get some background on why the sharing is done, the receiving user should be able to easily establish communication with the user who is sharing the content. The chat service is one of the prime candidates for this. See also the guidelines provided in section 3.6.6.2.

3.6.2.3 Video call

Please refer to section 3.9.2.2.

3.6.2.4 File Transfer

Since from a UX perspective File Transfer and Image Share are very similar services, content sharing without a voice call is limited to the sharing of videos. For sharing files or images when there is no voice call the File Transfer service as described in section 3.5 is used.

3.6.3 High Level Requirements

3-6-1 Content can be shared while on a CS or PS Voice call, thus it must be possible to have a voice and a data stream running simultaneously.

3-6-2 Each time a voice call is established, the user shall be offered the possibility to share content whenever possible

3-6-3 It shall be possible to establish a Content Sharing Session without an accompanying CS and PS Voice call.

3-6-4 It shall be possible to stream audio, along with video, during a Content Sharing session without a CS and PS Voice call.

3-6-5 Content Sharing shall be unidirectional. During a single content sharing session, the originator of the content sharing session can share content with the terminating party, but the terminating party cannot share content with the originator in the same session.

3-6-6 When sharing during a call, the receiving party may be offered the possibility to establish a session in the other direction when circumstances allow.

3-6-7 For content sharing without a voice call only one session may be established at a time. That is simultaneous sessions (regardless of the direction) are not allowed.

3-6-8 The content sharing service can be initiated by either end point involved in the voice call (e.g. the caller or the receiver). When a user initiates content sharing, an invitation is automatically sent to the other contact, which may be accepted or rejected. An acceptance shall stand for all the contents shared during the call.

3-6-9 End of communication (case of content sharing while on a voice call) shall be handled as follows:

- Content sharing session termination shall not lead to voice termination
- Voice call termination shall automatically terminate the content sharing session

3-6-10 The receiver shall have the possibility to save the shared content on his/her device if allowed by the sender

3-6-11 It shall be possible to assign a Service Provider configurable maximum content size allowed to be sent in an Image Share session. This enables the Service Provider of the inviting user's RCS client to control the maximum size of the content that the inviting user's RCS client is authorized to send in an Image Share session. The limitation should be transparent to the end-user.

3-6-12 It shall be possible to assign a Service Provider configurable maximum duration time allowed for a Video Share session. This enables the Service Provider of the inviting user's RCS client to control the maximum duration time of a Video Share session that the inviting user's RCS client is authorized to handle the limitation should be transparent to the end-user.

3-6-13 Shared content can be video and still images.

3-6-14 When a content sharing session is set up all of the recipient's devices shall alert the user

3-6-15 The recipient shall decide on which device they accept the call or session

3-6-16 When a call or a session is accepted or rejected from one device the pending invitation(s) shall be cancelled on all other devices that the recipient has.

3-6-17 When a call or a session is cancelled by the calling device, it shall be cancelled in all devices that the recipient has.

3-6-18 It shall be possible for a terminating party or an originating party to terminate the Content Sharing session.

3.6.4 Technical Realization

3.6.4.1 Video Share

3.6.4.1.1 Video Share during a voice call

Video Share during a voice call shall follow [PRD-IR.74] and take into account the handling of service capabilities and OPTIONS queries defined in sections 2.6.4.1 and 2.6 respectively. Furthermore to allow the user to accept the sharing on any device a Broadband Access client (see section 2.9.1.4) shall not automatically reject the INVITE request if it is not in a voice call with the sender. It shall therefore alert the user as if it was and handle the user's response as specified in section 2.11.

Interworking with Video Share terminals based on legacy specifications (i.e. the "already deployed terminals" option in [PRD-IR.74]) is not applicable in RCS.

3.6.4.1.2 Video Share without a voice call

A detailed description of this feature can be found in [PRD-IR.84] in sections 2.6.1 and 2.6.1.1 of that document. Additional information can be found in Sections 2.3.2 and 2.5.3 of [PRD-IR.84].

This means that for most cases the flows for setting up a session are very similar for the sharing with or without a voice call. The main difference for Video Share is that in the latter case the INVITE request includes instead of the complementary services feature tag ("+*g.3gpp.cs-voice*"), the ICSI feature tag with the MMTel ICSI and the IARI feature tag with the Video Share phase 2 IARI. That is respectively *+g.3gpp.icsi-ref="urn%3Aurn-7:%3A3gpp-service.ims.icsi.mmtel"* and *+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.gsma-vs"*.

The SIP *603 Decline* response shall be used for the automatic rejection of the incoming Video Share invitation without a voice call in case the initiator is included in the device's local blacklist that is described in section 3.6.1.3.

3.6.4.1.3 Content Share Recording

A new SDP attribute in the media level is defined to be used by the Video or Image Share sender to indicate, in the SIP INVITE, to the recipient RCS client that the shared media can be recorded/saved.

The new SDP attribute as defined in [IETF-DRAFT-SIPREC-PROTOCOL]:
`a=recordpref-attr = "a=recordpref:" pref where pref is set to "nopreference"`

An SDP example is *a=recordpref:nopreference*

If the shared media in a Video or Image Share session is allowed (determined by the sender) to be recorded/saved, the sender RCS client should include the above SDP attribute in the SIP INVITE toward the recipient when setting up the Video or Image Share session. If the shared media in a Video or Image Share session shall not be saved by the recipient RCS client the sender RCS client shall not include the above SDP attribute in the SIP INVITE.

A Service Provider can provision its RCS clients to not always include this SDP attribute in the SIP INVITE setting up the Video or Image Share session so the shared media will not be recorded/saved by the recipient RCS client.

If the new SDP attribute is included in the SIP INVITE setting up the Video or Image Share session, it is to the decision of the recipient RCS client (under the instruction of the recipient user or user preference) to determine if the shared media will be recorded/saved.

3.6.4.1.4 Video interoperability and encoding requirements

As presented in section 2.6.4.1, the Video Share service availability is mainly dependent on the network coverage. This is based on the assumption that both ends (source and destination) share the ability of handling a common video format and specific profile.

To guarantee the interoperability of RCS clients during Video Share scenarios, all RCS devices supporting the Video Share service (during or without a call) shall, at least, support the following video format:

- Video format: H.264/MPEG-4 (Moving Pictures Experts Group) Part 10 // AVC (Advanced Video Codec)
- H.264 Profile: Baseline Profile (BP)
- H.264 Level: 1b[30]

  NOTE: Please note that including this, it is highly recommended to support also the H.263-2000 codec with profile 0 Level 45 which is mandatory in RCS Release 1-4 Video Share that is based on [PRD-IR.74].

Next to these mandatory codecs, it is recommended to support additional video formats providing different levels of quality and to use them in an adaptive fashion depending both on the terminal status and the network conditions/coverage. As specified [RFC3264], formats must be listed in order of preference in the SDP media description. As such, additional codecs providing better quality than these mandatory ones should be listed in the SDP before the mandatory codecs. In any case for the encoding of the actual stream should be adapted to the currently available bandwidth and might therefore use bitrates lower than the maximum negotiated during session setup. To support this RTCP Receiver Reports (RR) shall be sent at least at a rate of one RR per second.

Note that in H.264, support of a certain level implicitly requires support for all lower levels, so a client supporting other H.264 levels should only indicate the highest level per profile that it supports.

Should an RCS terminal support several profiles, the final choice should be based on the outcome of the SDP media negotiation where both ends (sender and receiver) will present the supported video formats at that particular point (that is taking into account each device and network/connectivity status).

RTP payload handling shall be as described in section 7.4.3 of [3GPP TS 26.114] for the H.264 (AVC) video codec.

The receiving clients should preserve the aspect ratio of the incoming video stream, avoiding that video is stretched to fit the UI. The sending client may redefine the aspect ratio when supporting a flexible handling interface that could alternate between landscape and portrait (e.g. from 4:3 to 3:4 after the sending device has been rotated).

The originator of the Video Share session can indicate support for both Baseline (BP) and Constrained Baseline (CBP) Profiles.

The originator shall never use Flexible Macroblock Ordering (FMO), Arbitrary Slice Ordering (ASO), or Redundant Slices (RS) features of the profile no matter what the receiving client selects.

[30] The H.264 baseline profile 1b shall be encoded using the profile-level-id set to 0x42900B and the H.264 Constrained Baseline Profile 1b is 0x42D00B

When a receiving client that supports both CBP and BP receives the combination of BP and CBP profiles within the same SDP offer it shall select CBP profile when the CBP media format is listed first in the SDP m-line.

When the SDP negotiation results in the use of the Baseline Profile (BP), a client shall not send Single-Time Aggregation Packet type A (STAP-A) packets, even when the packetization-mode 1 has been negotiated. When accepting the use of the Constrained Baseline Profile (CBP) a client shall support the use of STAP-A packets when packetization-mode 1 was negotiated.

```
v=0
o=- 1323909835 1323909838 IN IP4 x.x.x.x
s=-
c=IN IP4 x.x.x.x
t=0 0
m=video 4284 RTP/AVP 118 119
a=sendrecv
a=rtpmap:118 H264/90000
a=fmtp:118 packetization-mode=1;profile-level-id=42d00b
a=rtpmap:119 H264/90000
a=fmtp:119 packetization-mode=1;profile-level-id=42900b
```

Table 68: Example of a VideoShare SDP Offer with both CBP and BP profiles, each using Level 1b, preference for CBP since it is first in m-line

3.6.4.1.5 Video interoperability in LTE/HSPA

Video Share used over high bandwidth connections such as LTE or HSPA allows high bitrate bearers, thus allowing better user experience e.g. when using a large screen.

As specified in [PRD-IR.74] and [PRD-IR.84], an RCS device shall support the H.264 video codec with baseline (and optionally Constrained Baseline Profile) profile and level 1.3[31] to provide 768 kilobits per second (kbps) video over an LTE bearer or over a similar high bitrate bearer. Please note that this codec is considered in addition to the mandatory formats specified in section 3.6.4.1.4.

If a second Video Share session is established in parallel, the H.264 video codec with baseline profile (and optionally Constrained Baseline Profile) and level 1.2[32] shall be used instead. The assumption for the use of a high bitrate bearer is that the connectivity and video parts of both terminals support it and have LTE or another high bitrate broadband access; otherwise the video bitrate will be reduced to the level 1b (as presented in section 3.6.4.1.4) to assure compatibility.

Also in this case the encoding of the actual stream should be adapted to the currently available bandwidth and might therefore use bitrates lower than the maximum negotiated during session setup.

3.6.4.1.6 Video Share duration

A configurable parameter allows the Service Provider to set the maximum duration of a Video Share session (see VS MAX DURATION in section A.1.5) in the UE. When one of the UEs which are sharing a live video stream detects that the maximum duration is reached, it

[31] The H.264 baseline profile 13 shall be encoded using the profile-level-id set to 0x42800D. For H.264 CBP level 1.3 it is 0x42C00D.

[32] The H.264 baseline profile 12 shall be encoded using the profile-level-id set to 0x42800C. For H.264 CBP level 1.2 it is 0x42C00C.

shall tear down the Video Share session by sending a SIP BYE request. When sharing a live video stream, if the sharing duration (send or receive) is approaching the duration limitation, a warning notification could be displayed for prompting the two UEs. When sharing a stored video, if the UE detects that the video file being shared exceeds the Service Provider's configured maximum duration, it shall either not set up the session or tear it down depending on whether it is the initiator or the receiver.

3.6.4.2 Image Share

Image Share during a voice call shall follow [PRD-IR.79] where the SIP OPTIONS query shall be handled as specified in section 2.6 of this document. Furthermore to allow the user to accept the sharing on any device a broadband access client (see section 2.9.1.4) shall not automatically reject the SIP INVITE request if it is not in a voice call with the sender. It shall alert the user and handle the user's response as specified in section 2.11.

To ensure that the request is sent to all devices with equal priority, clients using a PS voice service as defined in section 3.8 shall include the *+g.3gpp.cs-voice* feature tag in the Accept-Contact and Contact headers of the SIP INVITE request for content sharing. Unlike what is indicated in section 3.2 of [PRD-IR.79], those clients shall not include the MMTEL ICSI *urn:urn-7:3gpp-service.ims.icsi.mmtel* in these requests. This behaviour is in line with the other sections of [PRD-IR.79]. As described in [PRD-IR.79] the Image Share IARI is also included in the Accept-Contact and Contact headers (see Table 22 in section 2.6.1.1.2).

If the UE detects that the file being transferred exceeds the Service Provider configured maximum size (see IS MAX SIZE in section A.1.5), it shall either not set up the session or tear it down depending on whether it is the initiator or the receiver.

Note that all RCS services using MSRP, including Image Share, shall align with MSRP usage as described in section 2.8.

Details for image format as specified in [3GPP TS 26.141] will be followed.

3.6.4.3 Flows

3.6.4.3.1 General assumptions

In the following sections we will show the relevant message flows and reference UX. Please note that the following assumptions have been made:

- For simplicity, the internal mobile network interactions are omitted in the diagrams shown in the following sections.
- The terminal and the network support 2G DTM and it is therefore always possible to gracefully terminate the content sharing session related to a voice call provided the terminal remains switched on. If 2G DTM is not supported, the case where on one of the ends a handover occurs to 2G would be result in behaviour towards the other end and the network that is equivalent to the one described for the case of a client error.
- The device is in coverage supporting bidirectional Video Share (see section 2.7). If this were not the case, additional capability exchanges would be required when starting and terminating a sharing session to indicate respectively that the device cannot handle an incoming Video Share session and that it can handle such an incoming Video Share session again.
- The terminal comes with a front and rear camera. If one or both are missing, the user should be notified only with the available options.
- Prior to a voice call, the user accessed the client's address book, call log or dial-pad to make the call. As described in section 2.6, while these actions are performed a capability query is executed to double-check on the available capabilities. As in older RCS versions including in some non RCS clients, Video and Image Share services are only available in a call, an OPTIONS exchange is required once the call is established to

check on the capabilities during a call. This exchange can be initiated by either the sender or the receiver. In the following diagrams it is assumed that this initial exchange (OPTIONS and response) has already taken place, and therefore, both ends are aware of the capabilities and the available RCS services.

- In the diagrams it is assumed for simplicity that MSRP chunking is enabled. This is for representation purposes and it is up to the OEM to decide whether MSRP chunking is enabled or not.
- The flows in Figure 88, Figure 89, Figure 90, Figure 91, Figure 92, Figure 93, Figure 94, Figure 96, Figure 97 and Figure 98 show an OPTIONS exchange at the end of the flow. If the capability exchange is done using Presence the equivalent Presence mechanism will be used.

As shown in section 3.6.6, the different entry points for sharing without a call lead to different screens. Therefore for consistency the flows highlight the slightly more complex case of sharing during a voice call.

3.6.4.3.2 Exchange capabilities during a call

The assumptions in this case are that User A and B are on a call when the capabilities of one of the users change (due to a hand-over to a different data carrier for instance). Therefore the other end has to be informed using the OPTIONS message[33]

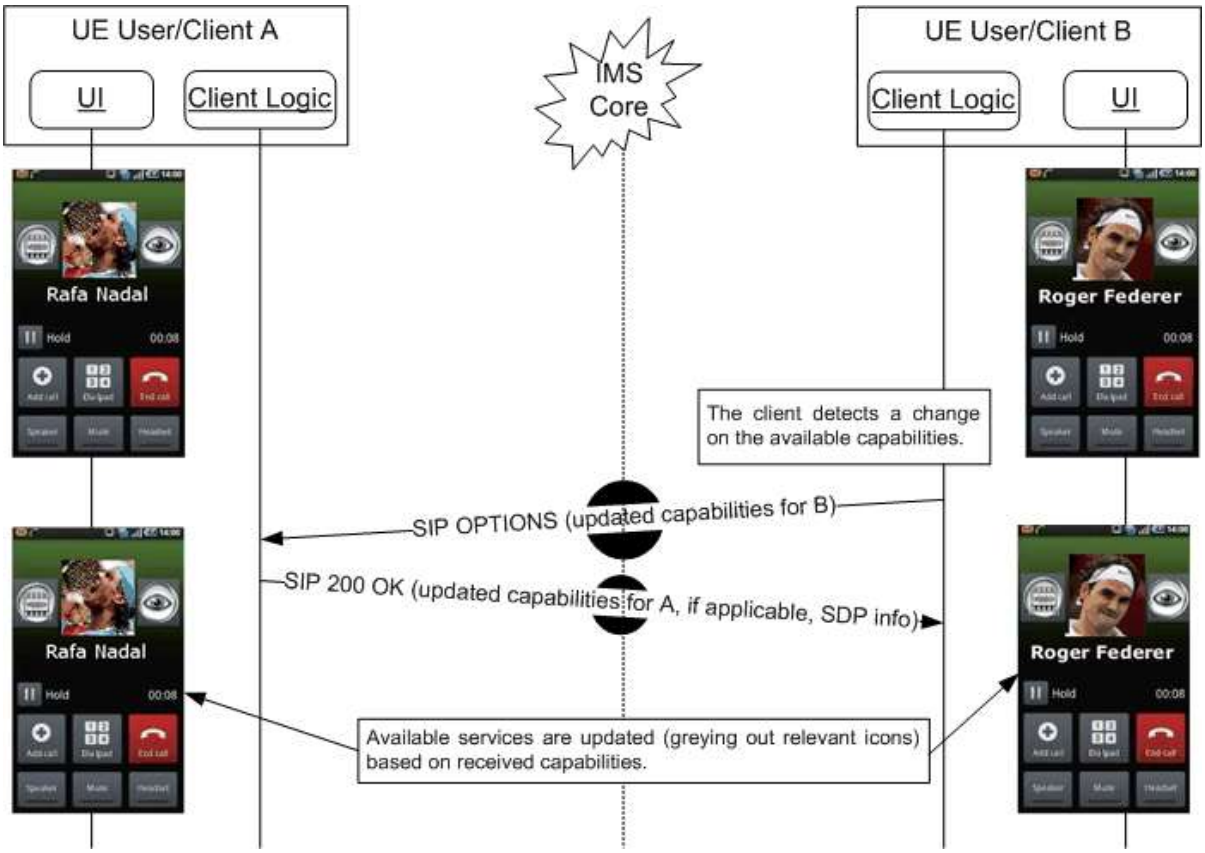


Figure 86: Capabilities exchange during a call

[33] The SDP information included in the response to the OPTIONS request is required due to the compliancy to [PRD-IR.74]. This will only be used during OPTIONS exchanges related to a call. The Video Share service shall only be considered to be available if at least one codec in the received SDP is supported by the client.

3.6.4.3.3 Share video

The assumptions in this case are that both User A (wanting to share video) and User B (recipient wanting to receive it), have successfully performed the capability query, either as shown in section 3.6.4.3.2 or as specified in section 2.6 depending on whether or not the sharing is done in relation to a voice call. Therefore, both clients are aware that video sharing is possible (both UEs on a 3G+ or Wi-Fi).

In this case RTP is the protocol used to stream the video data, so it can be reproduced in real-time on the other end.

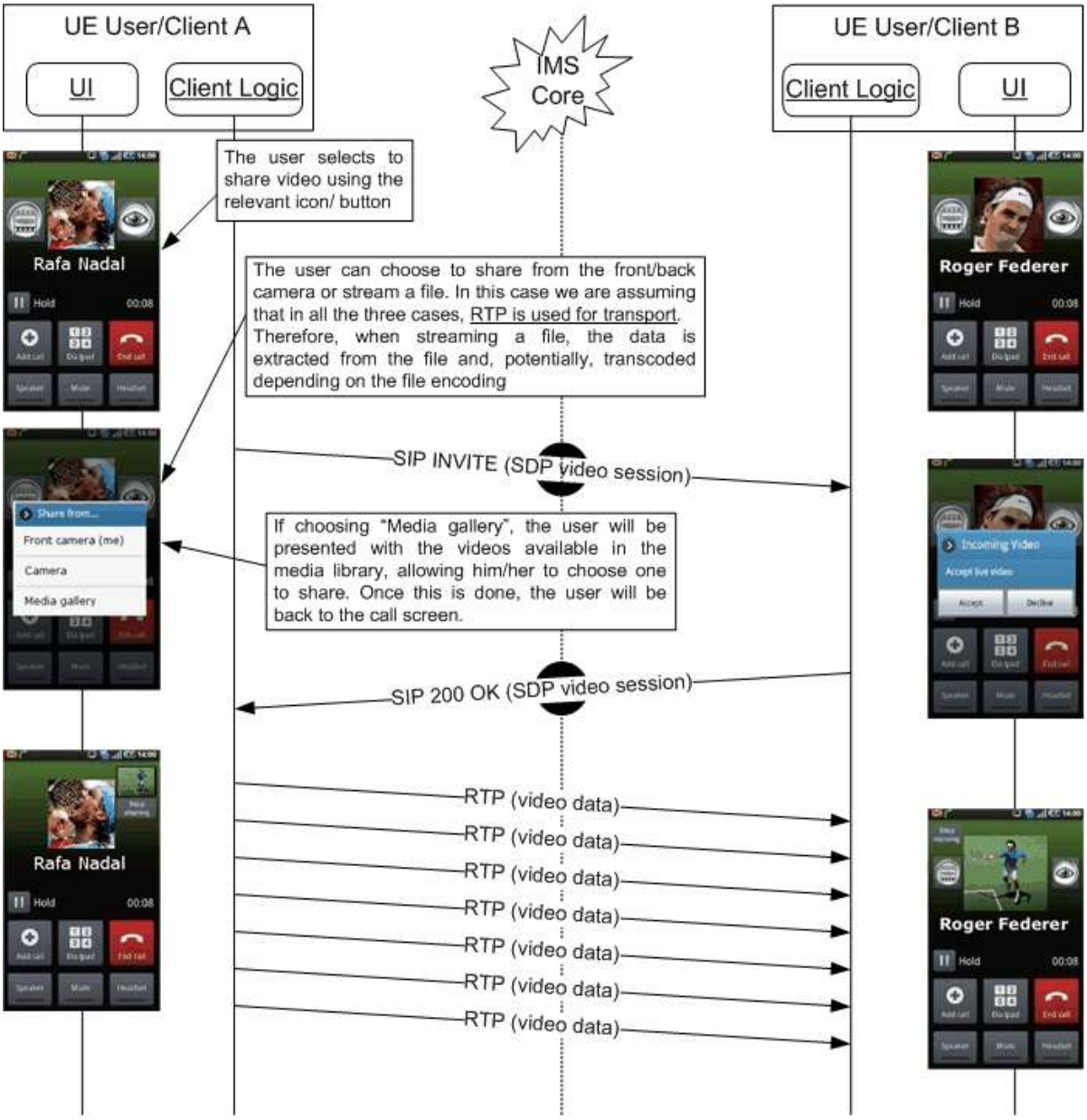


Figure 87: Share Video

3.6.4.3.4 Stop sharing video (RTP): Sender initiated

The assumptions in this case are that User A is sharing a video (through RTP) with User B. However User A no longer wants to keep sharing it.

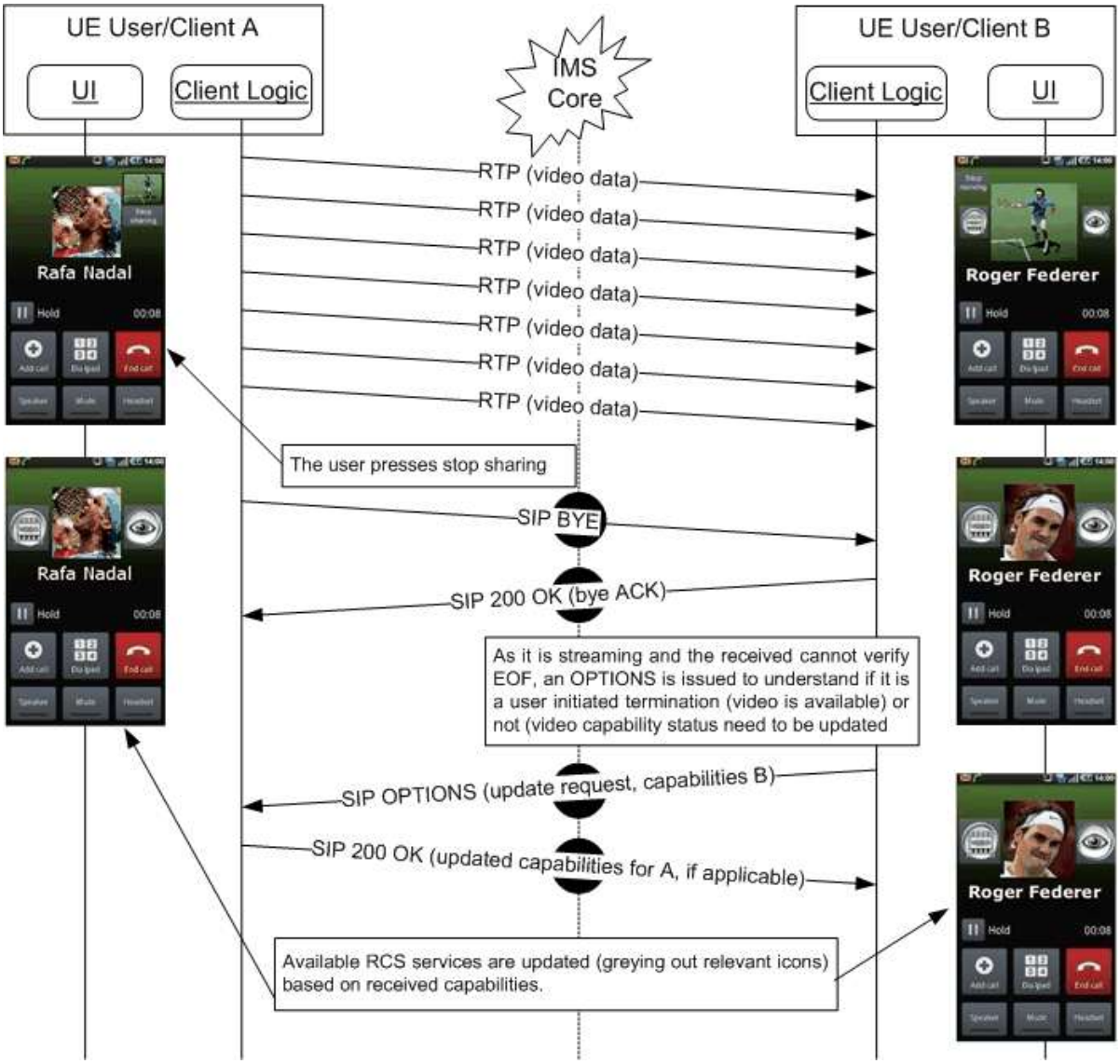


Figure 88: Sender stops sharing video

NOTE: in case of sharing without a voice call the OPTIONS exchange at the end of the flow is not applicable as for all other services it can be assumed that the other party would be informed if a service is no longer available.

3.6.4.3.5 Stop sharing video (RTP): Receiver initiated

This case is equivalent to the previous one. However, it is the receiver (User B) who does not want to keep receiving the video.

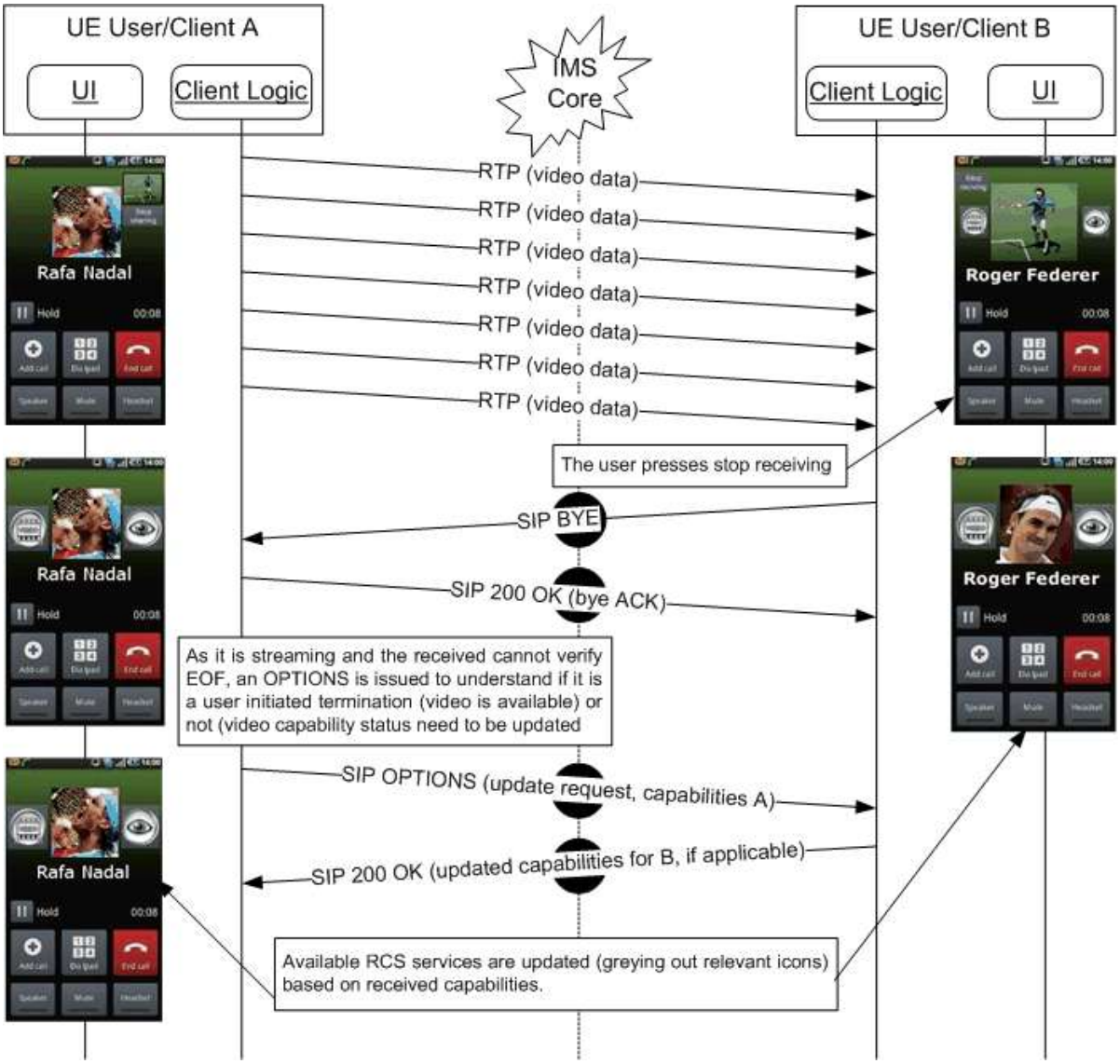


Figure 89: Receiver wants no longer to receive video

NOTE: in case of sharing without a voice call the OPTIONS exchange at the end of the flow is not applicable as for all other services it can be assumed that the other party would be informed if a service is no longer available.

3.6.4.3.6 Stop sharing video (RTP) as the required capability is no longer available

The assumptions in this case are that User A is sharing video (RTP) with User B, and either User A or User B is no longer capable (for instance because the terminal is busy, suddenly has no LTE, 3G+ or Wi-Fi coverage available without triggering an IP reconfiguration or loss of connection) of sending or receiving a video. Please note that in the example, it is assumed that the sender (User A) is the one losing the capability. This sequence will be equivalent if:

- The receiver (User B) loses the capability to receive video: The BYE and OPTIONS exchange would be initiated by the receiver (User B) in this case.

- Both lose the capability to share video: The BYE and OPTIONS exchange message would be initiated by the client that is the first one to lose the capability in this case.

In losing the capability to send video, the case in which there is an IP reconfiguration is excluded. Please note that this particular case is covered under the “Client Error” section later in this section (see 3.6.4.3.12 and 3.6.4.3.13).

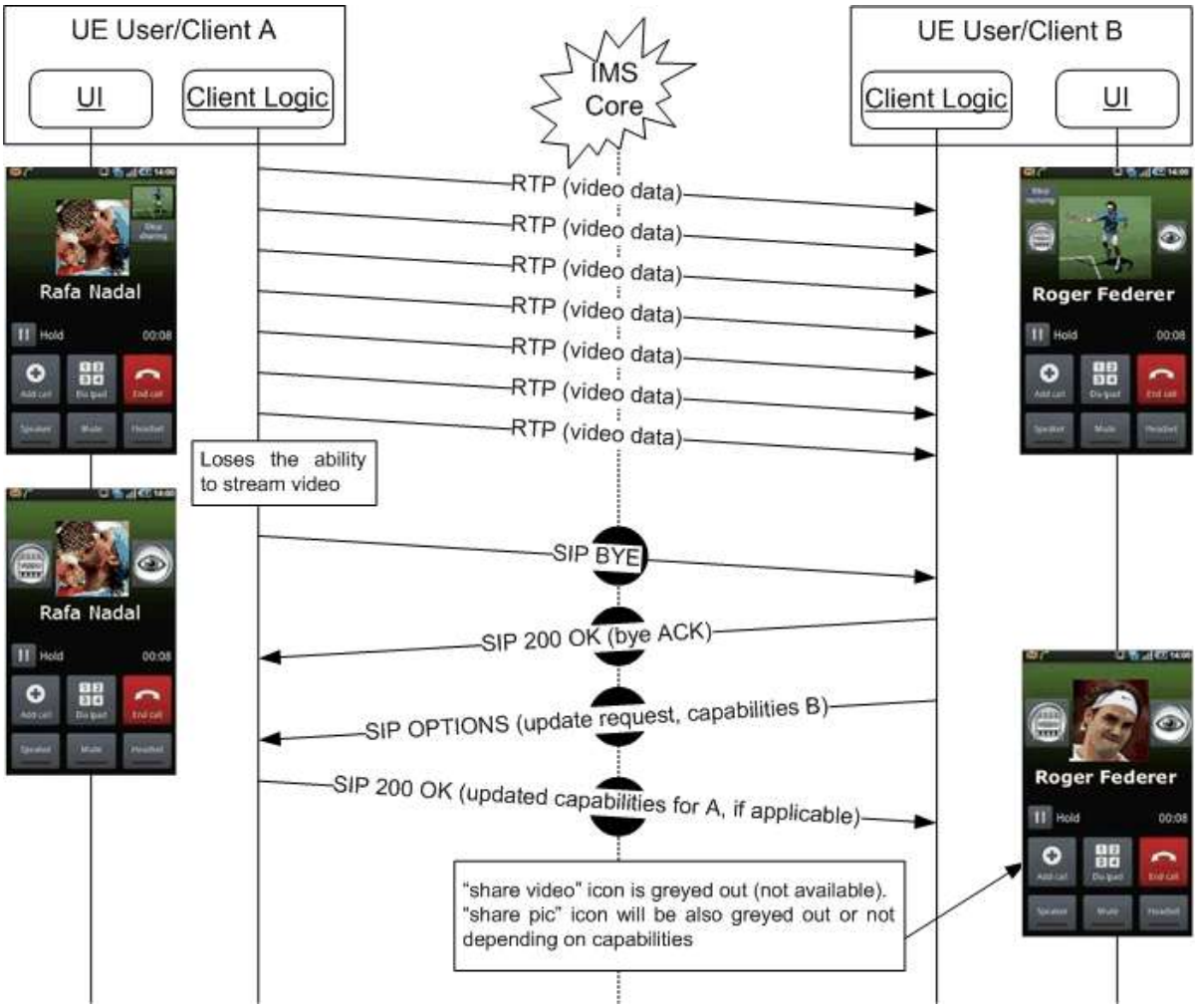


Figure 90: Video can no longer be shared (capability not available)

3.6.4.3.7 Share pictures during a call

The assumptions in this case are that both User A (wanting to share picture) and User B (recipient wanting to receive it), have successfully exchanged the OPTIONS messages. Therefore both clients are aware that Image Share is possible (that is both UEs are on an LTE, 3G+ or Wi-Fi network).

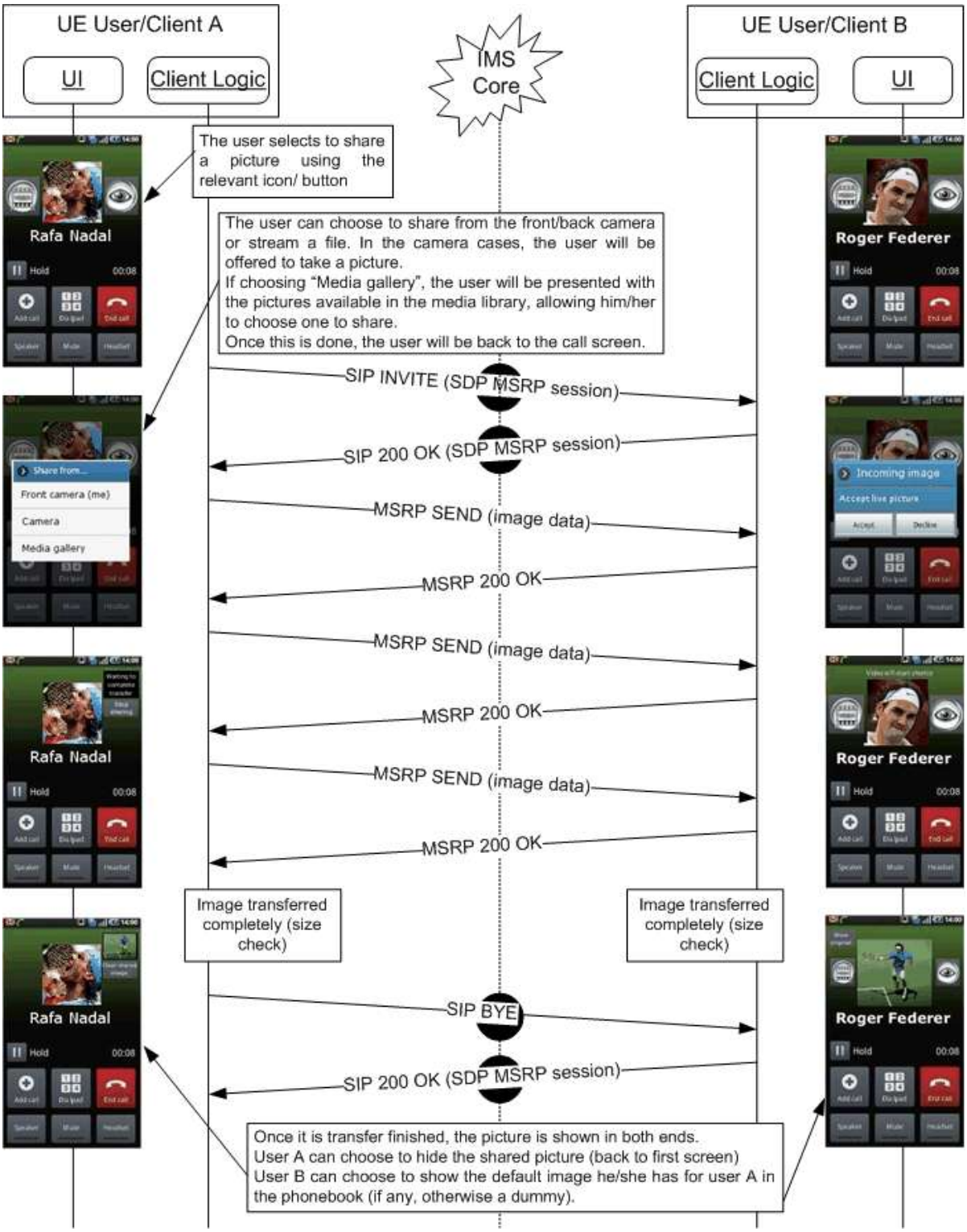


Figure 91: Sharing a picture during a call

3.6.4.3.8 Stop sharing a picture during a call: Sender initiated

The assumptions in this case are that User A is sharing a picture with User B, the transfer is still ongoing, but User A no longer wants to keep sharing the picture.

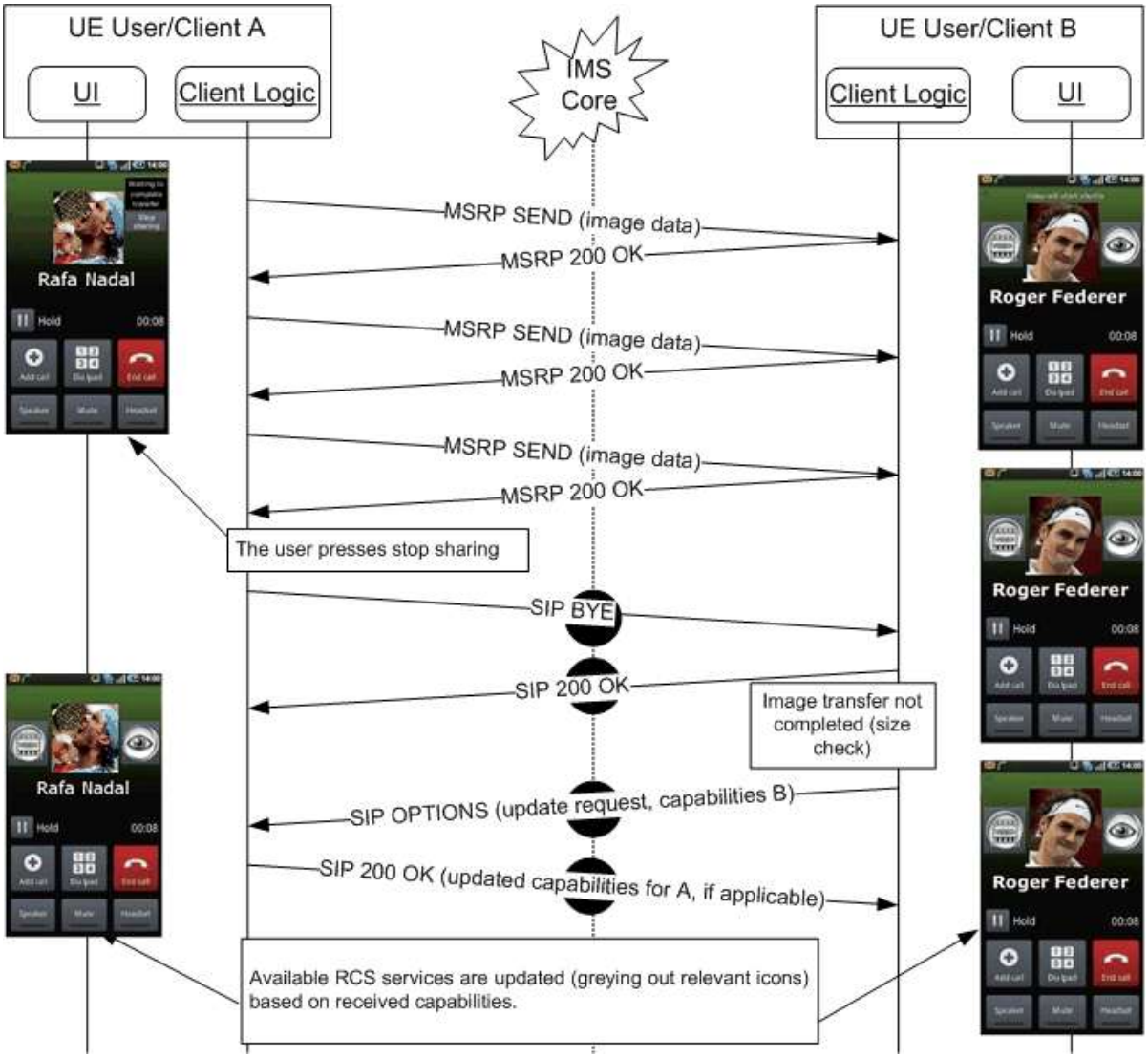


Figure 92: Sender stops sharing a picture during a call

3.6.4.3.9 Stop sharing a picture during a call: Receiver initiated

This case is equivalent to the previous one. It is however the receiver (User B) who does not want to keep receiving the picture.

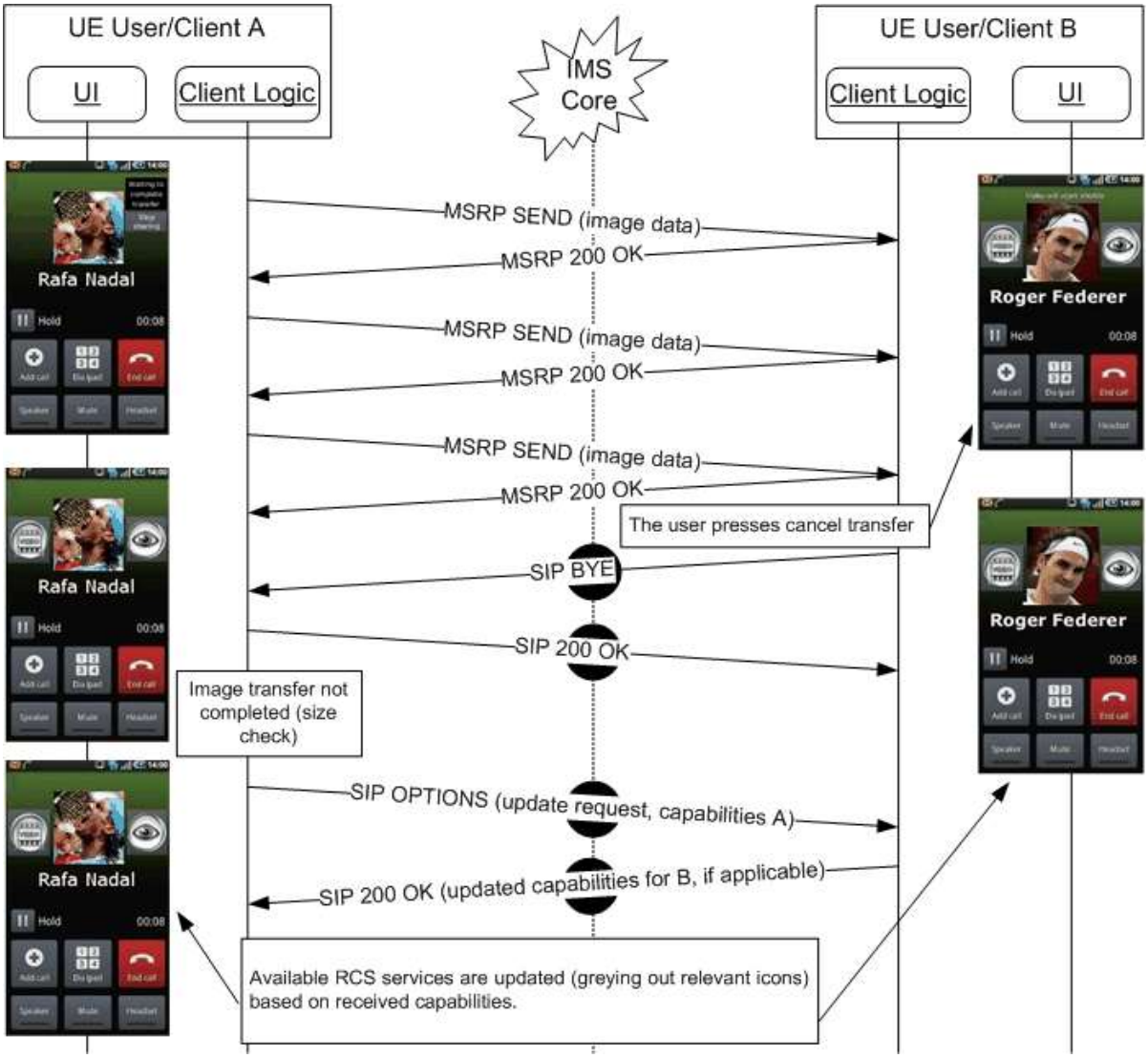


Figure 93: Receiver stops picture sharing

3.6.4.3.10 Stop sharing a picture during a call as the required capability is no longer available

The assumptions in this case are that User A is sharing a picture with User B, the transfer has not yet finished, and either User A or User B are no longer capable (for instance because the terminal is busy) to sharing or receiving the image respectively. Please note that in the example it is assumed that the sender (User A) is the client losing the capability. The sequence will be equivalent however for:

- The Receiver (User B) losing the capability to receive pictures: The BYE and OPTIONS exchange would be initiated by the receiving client (User B) in this case.
- Both lose the capability to share pictures: The BYE and OPTIONS exchange would be initiated by the first client to lose the capability in this case.

Please note that there is an exception to stop a file transfer due to capabilities. If one of the users is left with 2G coverage (on a DTM terminal) once a transfer has started, the transfer may continue until completed, provided the handover did not trigger an IP bearer

reconfiguration. Once the transfer is completed however, picture sharing will no longer be available as a service during the call.

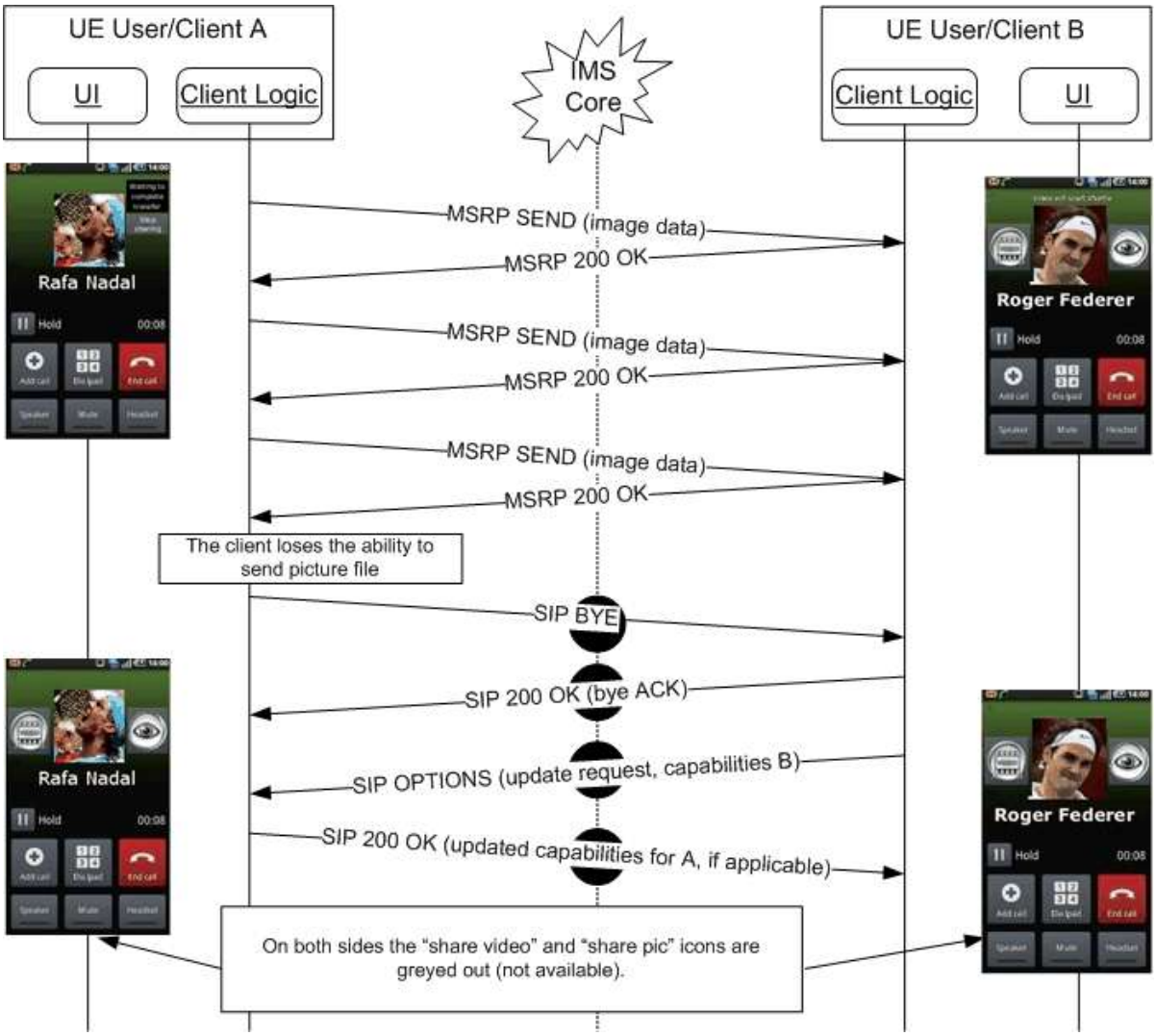


Figure 94: A picture can no longer be shared during a call (capability not available)

3.6.4.3.11 Decline share video or picture

User A wants to share a video or picture with User B. User B however does not want to receive it. Please note that it is assumed that both Video and Image Share is possible (that is the proper capabilities are available).

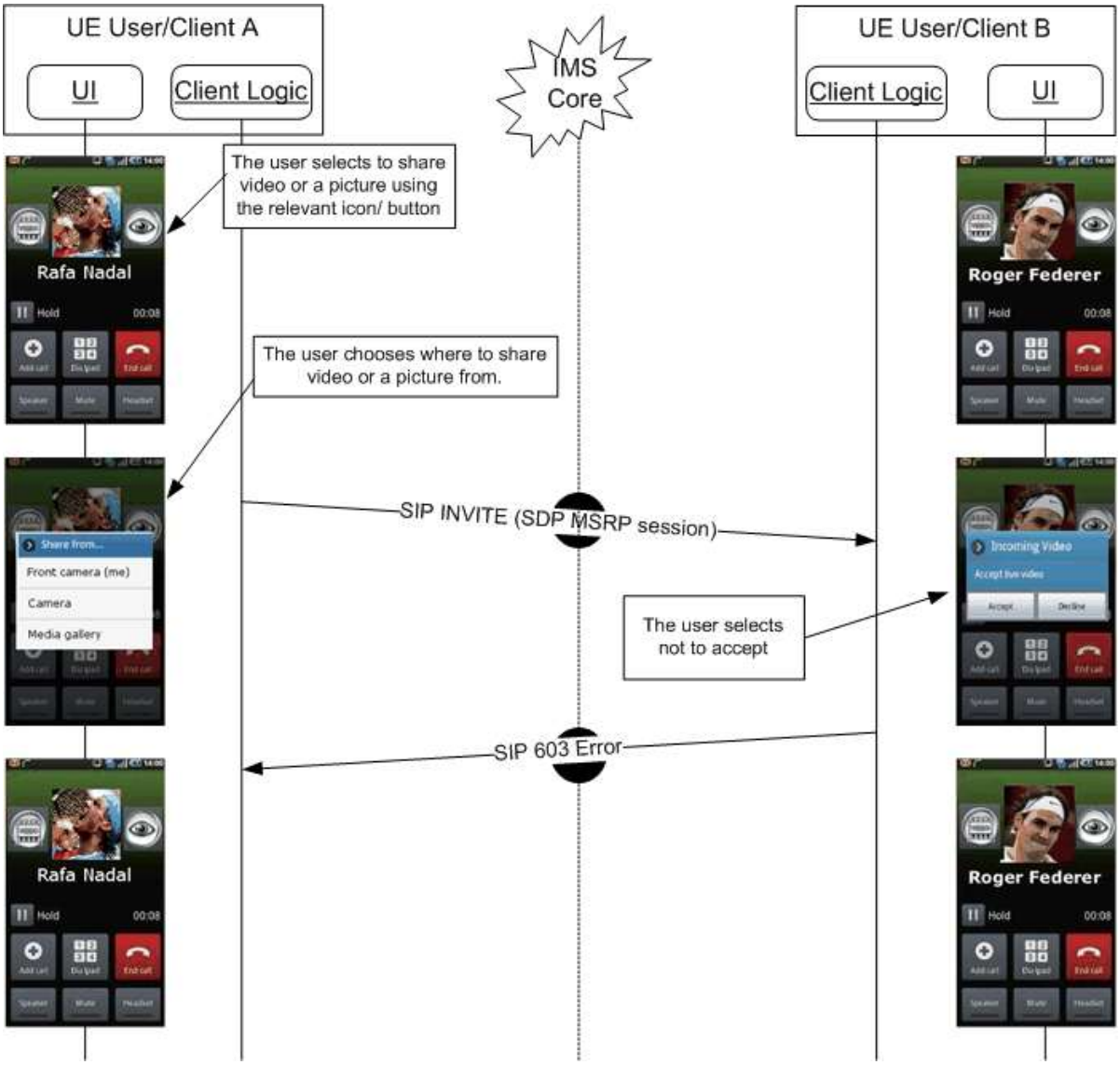


Figure 95: User declines sharing a picture during a call

3.6.4.3.12 Non-graceful termination (sender): Video or picture sharing

In this case, User A is sharing video or a picture with User B. Suddenly, User A's connection to the network fails (This may for instance be due to a client error, a reboot of the device, the loss of the data bearer, a switch in data carrier [for instance 3G+ to 3G] causes an IP layer reconfiguration and so on).

In the following flow, it is assumed a video transfer (RTP) was taking place. It will be equivalent however to the case an MSRP transfer (Image Share or video sharing via File Transfer) was taking place and was not finished:

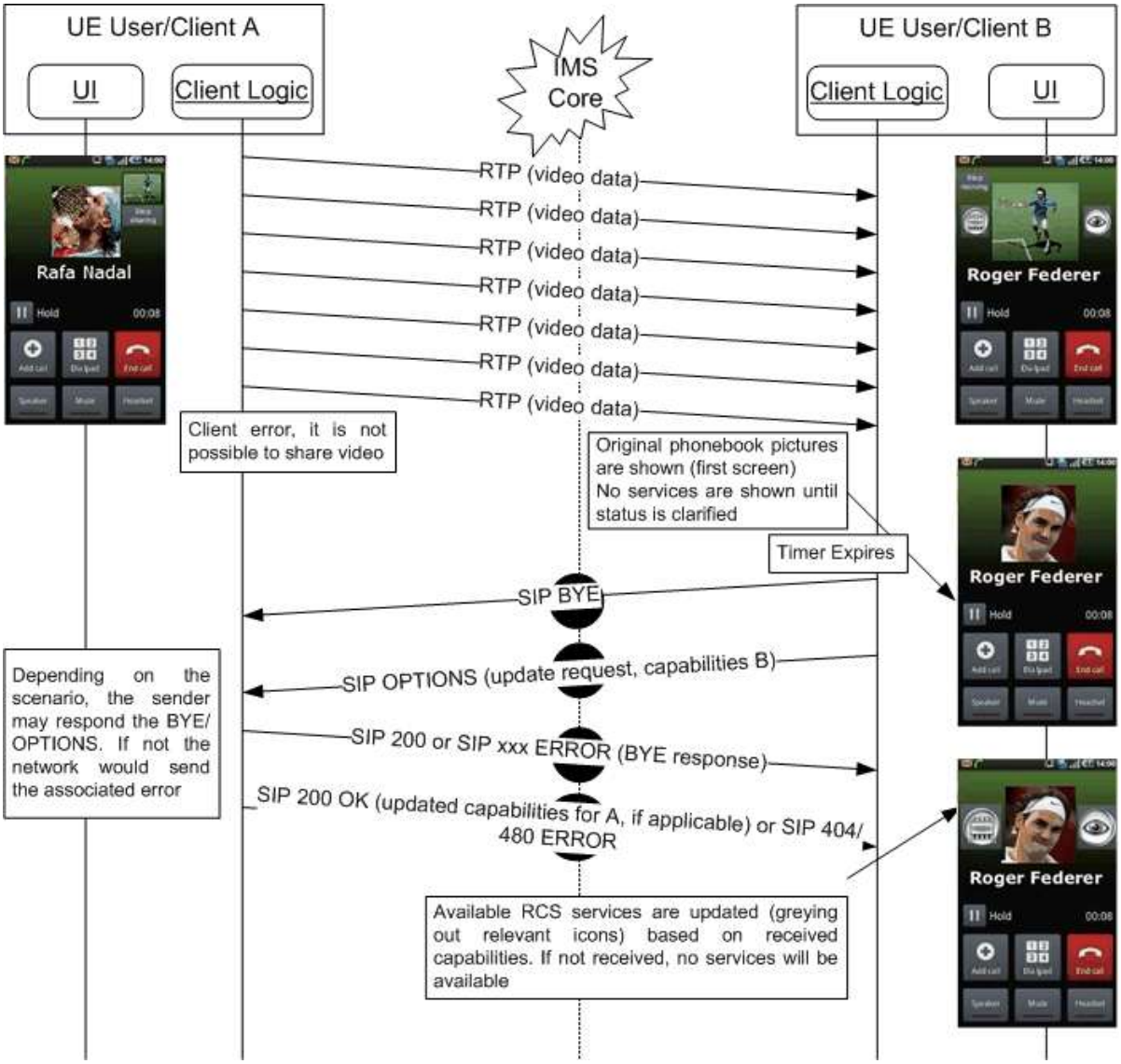


Figure 96: Non-graceful termination (sender) for video

3.6.4.3.13 Non-graceful termination (receiver): Video or picture sharing

To protect the IMS Core network from cases where both the sender and the receiver become unresponsive or unreachable before they had time to terminate the SIP session, the RCS Client shall use the procedure described in [RFC4028] in a similar way to the one mandated in [RCS5-SIMPLEIM-ENDORS], that is the RCS client initiating a SIP session must request the role of refresher and the option tag 'timer' must be included in a Supported header.

The Session-Expires and Min-SE values announced by an RCS client must be configurable by the Service Provider.

This use case is identical to the previous use case, except that in this instance User B (receiver) loses the ability to receive/process MSRP messages (this can for example be due to a client error, a reboot of the device, a loss of the data bearer and so on).

In the first flow diagram it is assumed that an Image Share transaction was taking place through MSRP:

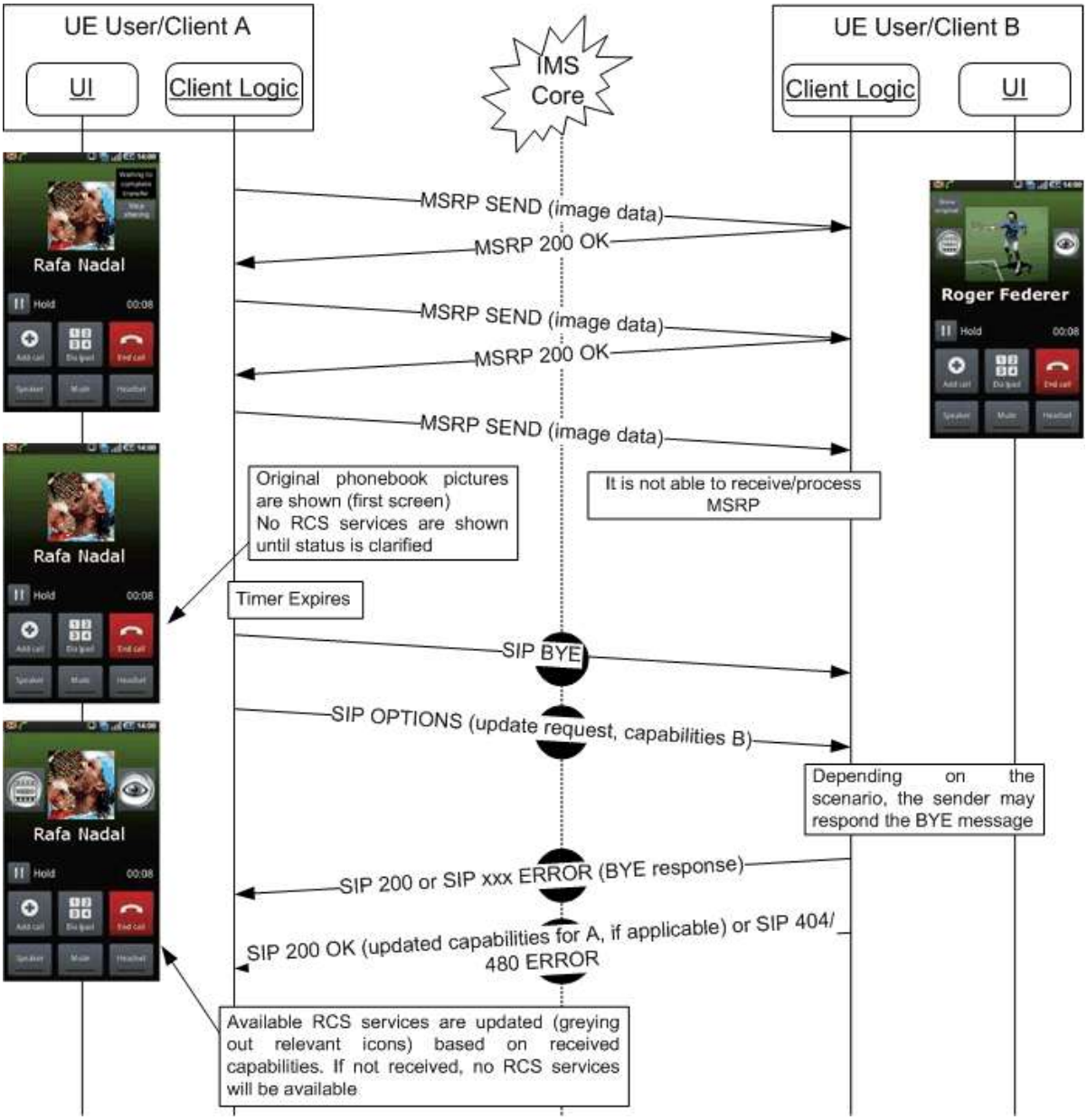


Figure 97: Non-graceful termination of picture sharing during a call

In the second flow it is assumed that a Video Share transaction was taking place through RTP:

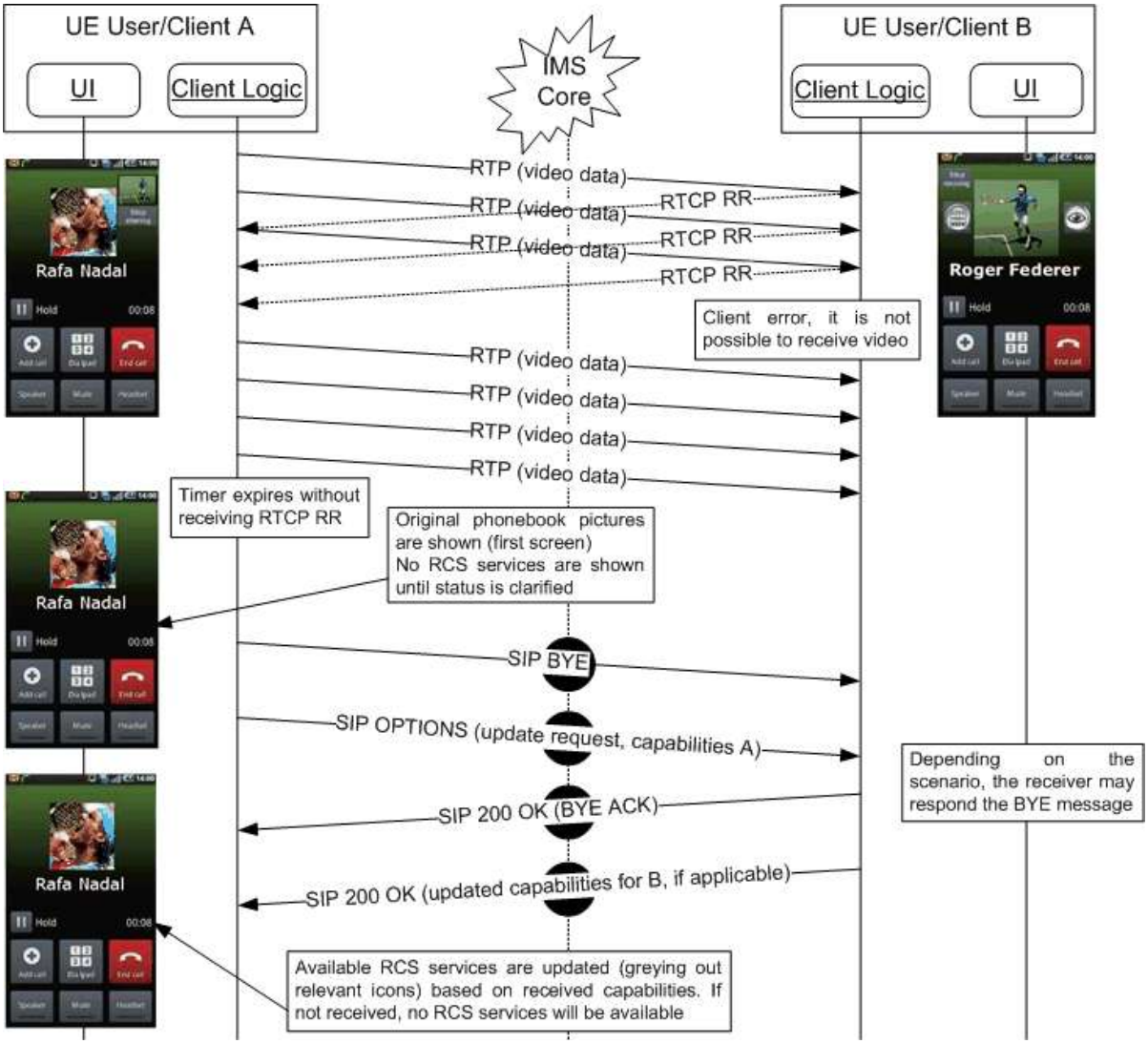


Figure 98: Non-graceful termination of video sharing during a call

3.6.5 NNI and IOT considerations

The NNI interfaces for content sharing services shall behave according to the procedures described in section 2.12 and referred documents.

3.6.6 Implementation guidelines and examples

3.6.6.1 Content Sharing during a call

As this is about sharing during a call, for both the sender and the receiver the sharing always starts from the call screen where the capabilities for sharing to the conversation partner in the voice call are shown. The user can then select one of the available services after which they will select the source of the sharing. A session will then be set up and the user will see the content that is being shared.

3.6.6.1.1 Video Share

The description above leads to following user experience for the initiator of a Video Share:

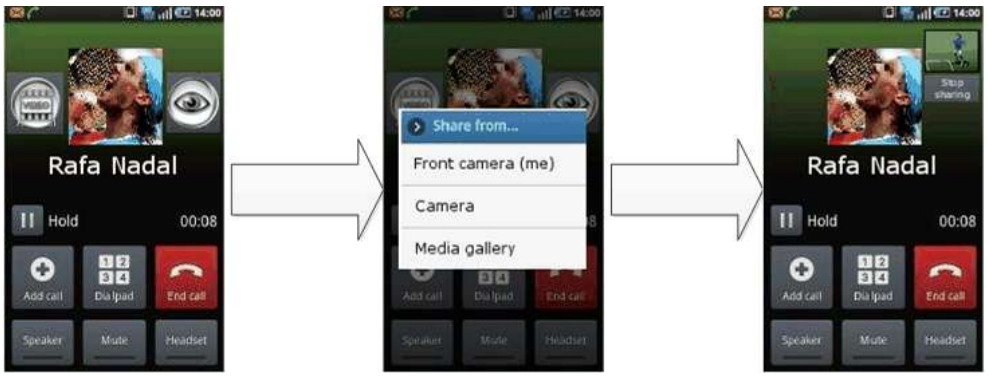


Figure 99: Reference UX for Video Share during a call (initiator)

A user invited for Video Share during a call first receives an additional invitation and if they accept, they are shown the video with the possibility to stop the sharing:

Figure 100: Reference UX for Video Share during a call (recipient)

NOTE: When the receiver accepts the sharing from the device that is involved in the voice call this acceptance applies automatically to all further sharing requests during that call.

3.6.6.1.2 Image Share

For Image Share the experience is similar than the one for Video Share shown in section 3.6.6.1.1. As it requires the transfer of a large file before something can be displayed rather than being able to stream immediately, there is a transfer delay. This leads to the following user experience for the sender:

Figure 101: Reference UX for Image Share during a call (sender)

A user invited for Image Share during a call first receives an additional invitation and if they accept, they are shown the image with the possibility to stop the transfer initially and stop displaying the image once transferred:

Figure 102: Reference UX for Image Share during a call (receiver)

NOTE: When the receiver accepts the sharing from the device that is involved in the voice call this acceptance applies automatically to all further sharing requests during that call.

3.6.6.2 Content Sharing without a call

From the UX, there are five possible entry points to these services:

1. Address book/Call-log: Content sharing can be initiated with any registered contact providing the right capabilities are in place – contact oriented initiation.
2. Media gallery/File browser: The user can browse, select a (media) file and then share with an RCS user – task contact oriented initiation. Only RCS capable users shall be displayed as candidate recipient of the sharing. Once video sharing is selected, the user will be presented with the complete list of RCS contacts (including contacts which are currently not registered).
   In this case, a capability exchange as defined in section 2.6.1 is performed once a contact is selected from the list.
3. Camera application: The experience is analogous to the media gallery/file browser experience with the difference that the user is able to only select the last video (and, in some cases, one video from the camera gallery) to be shared.
4. Chat window: From the Chat (one-to-one Chat only) window a video can be shared using the relevant button/icon. The experience is identical to the address book/call-log. The user is redirected to the media gallery or file explorer where he/she can choose a file or media content which then shared.
   The capability exchange as defined in section 2.6.1 is performed when the user opens up the menu in which the available content sharing options are offered
5. Call screen: They can share a video either by using the camera (front or back) or choosing a file from the media gallery. Please note this has been covered in detail in section 3.6.6.1.

When transferring a video whilst not in an existing session (i.e. when not in a call or Chat) and after the transfer has commenced (i.e. the user accepted the incoming file or content sharing session) the shared content is presented to the recipient in a Chat UX. This establishes a communication context for the transfer as the recipient may want to know why the sender is sharing the file. Please note that at the time the sharing is presented, the Chat session is not started; the Chat session will only start if and when the receiver sends a chat message back to the sender.

3.6.6.2.1 Video Share

For video transfer the behaviour follows the one described in section 3.6.6.2. Once the sharing commences, depending on the entry point the video is shown in a dedicated screen or within a screen applicable to the context. This leads to following handling for the different entry points

- Address book/Call-log: Following the address book interaction, the user can select the source of the content they would like to share (the media gallery or one of the available cameras). Once the sharing commences, the shared video is shown on the screen until the user terminates the sharing

Figure 103: Reference UX for accessing Video Share without a call from address book/call-log

- Media gallery/File browser: The user can browse, select a video to share with another RCS user.

Figure 104: Reference UX for accessing Video Share without a call from media gallery

- Camera application: The experience is analogous to the media gallery/file browser experience with the difference being that the user is able only to select sharing the live image captured by the camera (and, in some cases, the last video or a video from the camera gallery) to be shared.
- Chat window: From the Chat (one-to-one Chat only) window a video can be shared using the relevant button/icon. The experience is identical to the address book/call-log. The user is redirected to the media gallery where they can choose a video which is then shared or use the camera to share a live video.

Figure 105: Reference UX for accessing Video Share without a call from chat window

- Call screen: This is video sharing during a call and has been covered in detail in section 3.6.6.1.1

For the recipient the video sharing is handled through a chat user experience meant to provide a communication context to discuss the shared content if needed. The Chat session needed for that will only be established as soon as the recipient wants to send a message to the sender. This leads to following user experience:

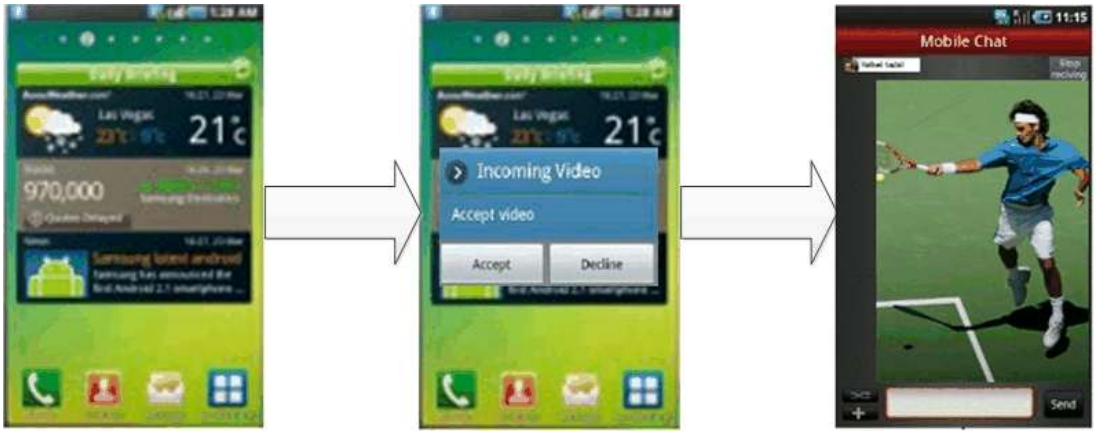


Figure 106: Reference UX for accessing Video Share without a call for the receiving side

If there was already an ongoing chat session when the invitation for Video Share came in, that session is reused for showing the video rather than starting a new one:

Figure 107: Reference UX for accessing Video Share without a call for the receiving side during a chat

3.7 Social Presence Information

3.7.1 Feature description

3.7.1.1 Social Presence definition

Social presence is seen as a piece of information for buddies to let them know about what you are doing, your mood, status, and so on. The user is given the possibility to publish personal data, which configures the users Social Presence Information, or “personal profile”.

As an illustration, the group of contacts with whom a presence relationship is established can be seen as the closest contacts of a certain user (friends, family, colleagues, and so on.).

Social Presence Information (included in the personal profile) does not replace the legacy contact's vCard in the address book of the user (for example. the contact name and other contact details shall not be impacted).

The Social Presence Information shall be controlled by the end user and easily configurable.

Having established a Social Presence Relationship with a certain contact, the Social Presence Information shall be visible from the Enhanced Address Book (EAB). It should also be visible from other places on the device, like for example the communications log, or message folders.

3.7.1.2 Service Fundamentals

In the EAB, the contact information is extended with social presence information and foresees the following attributes:

- Availability, indicates the user's (un)willingness to communicate,
- Portrait icon, depicting the user (e.g. a photo or image provided by the contact himself)
- Free text, including textual note and possibility to add emoticons (automatic translation of some specific characters into smileys)
- Favourite link, to publish hypertext link of personal and/or favourite site
- Timestamp, date of the last update of the profile, generated automatically.
- Geolocation, depicts the user location

The attributes Availability, Portrait icon and Favourite link are profiled from the standards bringing a new user experience.

The Availability allows a user to inform a contact that they are currently in a situation when it is possible/not possible to communicate.

The Availability is controlled fully by the user and not automatically switched on or off.

With the portrait icon, it is possible to publish a photo or an icon, which is shown in the EAB of the user's contacts. This is a new user experience while a user has full control of the portrait displayed at his contacts. Within RCS the size and dimension of the photo is specified.

The favourite link attribute allows sharing additional social presence information. Such a link can point to e.g. a blog.

With geolocation, two RCS users are able to see where they are located and share this information with each other.

Authorisation to share social presence is based on the symmetry principle.

If sharing of social presence is accepted after invitation, both parties will see each other's presence attributes. If social presence sharing is terminated by one of both parties, both parties will end seeing each other's social presence attributes.

When a social presence relationship with a contact is set up from one device (e.g. the broadband client on PC) this relationship will also be visible on the other devices of the user (e.g. a mobile device).

The RCS invitation experience is improved with a personalized invitation. For easy identification of invitations coming from contacts not yet registered in the user's address book, it is possible to define a nickname to be used in presence invitations.

By choosing whether or not the contact is a VIP contact (see section 3.7.1.4.9), it will be possible to choose for a contact with which social presence is shared whether updates to

that contact's social presence information should be reflected in (near) real time or whether those updates should be retrieved through some low frequency polling for them.

3.7.1.3 Social presence attributes

3.7.1.3.1 Availability status

A user will be able to set the state of Availability status (as part of Social Presence Information)

There are two possible states that can be selected by the user, from their RCS Client:

1. **State#1.** From the RCS Client, the user can set Availability status information as state#1. This state is informative and means that user is available and willing to communicate The way state#1 is displayed to the user is implementation dependent, and subject to own Service Provider policies.
2. **State#2.** From the RCS Client, the user can set Availability status information as state#2. This state is informative and means that the user is unavailable or not willing to communicate (e.g. busy) and will probably not respond to any incoming calls or messages. The way state#2 is displayed to the user is implementation dependent, and subject to own service provider policies.

These states are informative. When a user sets Availability status information as state#1 or state#2 from the RCS Client, the user still has the possibility to make outbound communications (e.g. calls/messages) and receive inbound communications (e.g. calls/messages).

The Availability status information has a permanent nature. It remains unchanged until the user decides to modify it (as state#1 or state#2) from their RCS client

The Availability status information is not linked with any particular user's network connectivity situation (e.g. temporary loss of network connectivity, device switched off).

The RCS device and the Presence Server shall support the availability status feature.

The Service Provider shall be able to choose to enable or disable use of Availability status feature, according to its own Service Provider policies.

The RCS device of a user whose Service Provider enables the use of the Availability status feature, will not receive any Availability status information associated with a presence-enriched contact, which subscribes to a Service Provider, which has disabled the use of the Availability status feature.

The RCS device of a user who's Service Provider disables the use of the Availability status feature, will not display, subject to Service Provider policies and bilateral agreements between Service Providers, any Availability status information associated with a presence enriched contact which subscribes to a Service Provider which enables use of Availability status feature. Moreover, an RCS device using a Service Provider that disabled the use of the availability status information shall not offer to the user the ability to set Availability status information.

3.7.1.3.2 Favourite Link

One of the attributes in the Social Presence Information allows the user to add or update one hypertext link, which (when selected) may redirect, for instance, to an extension of the user's Social Presence Information (for example a mobile blog).

The user shall be able to edit the hypertext link (expressed as a Uniform Resource Identifier as defined in [RFC2396]) via two modes:

1. Manual mode, where the user types in manually the URI
2. Automatic mode, where the user selects one URI from a predefined list.

The Service Provider shall be able to choose whether to offer its customers only manual mode, only automatic mode, manual and automatic modes, or no mode at all.

A clickable link is displayed in a *detailed view mode* of the Social Presence Information, where shared information about the user (portrait icon, free text and URI) can be seen in larger size than in the EAB itself (*list mode*).

When the user edits a new hypertext link, those contacts, which the user has established a Social Presence Relationship with, are notified, that is a visual change of value of favourite link attribute, for example when the user updates their portrait icon or free text.

When a user clicks on the link of a presence-enriched contact, the appropriate native handler for linked content (for example browser) shall be launched.

When the user closes the handler, they return automatically to the presence enriched contact's *detailed view mode* of the Social Presence Information, from where the handler was launched.

A revoked contact shall not be able to click on the hypertext link. However, please note that there are no restrictions that prevent the watcher from being able to save the URI in their browser and further access to this URI.

It is possible to display a "user friendly" label for the favourite link instead of the actual URI.

Instead of displaying the URI the RCS user can display a personal label. The maximum size of characters recommended is 20 (this can be set by Service Providers as a provisioning parameter). It shall not be larger than 200 characters.

3.7.1.3.3 Geolocation information

Geolocation information is a combination of declarative text always manually edited/updated by the user; and/or coordinate information (x, y) that is displayed on a map.

The maximum character size of declarative location text information the end-user can enter can be set by the Service Provider as a provisioning parameter. It shall not exceed 200 characters. The text information on the receiving part cannot exceed 200 characters and is not limited by any provisioning parameter.

Time Zones can be shared as part of geolocation information, allowing users to view what the local time is at their friend's location.

A provisioning parameter can be set in the network by Service Providers to control the maximum time the published location information will be considered to be valid (for example, one month).

The user must be able to delete his location information (empty text field, no position on map).

Location information must be interoperable between RCS clients no matter how users choose to update their information. For example, if User A has updated his location on a map (with x, y coordinates) and User B (authorized contact) is using RCS clients without a map feature (and only supporting declarative text), they must still be able to view User A's location as a intelligible text, using the declarative text information (if available), not as raw x, y information.

To avoid excessive traffic on the network due to very frequent location updates, it is recommended that a provisioning parameter can be set in the network to remotely set a minimum duration between updates sent from the client/device.

The geolocation feature can be provided on non-GPS (Global Positioning System) enabled devices.

3.7.1.4 Social Presence Authorization

RCS users shall feel confident in publishing their Social Presence Information, and be guaranteed that their privacy is respected. Therefore, mechanisms are defined below that allow users to accept/reject an invitation to establish a Social Presence Relationship, since this may imply sharing certain potentially private information, such as portrait icon or free text.

3.7.1.4.1 Social Presence Information sharing request principles

Reactive authorization shall be used, that is when User A invites User B to share Social Presence Information, User B receives an authorization request.

When receiving an invitation to share Social Presence Information from User A, User B can:

- **Accept** the invitation.
- **Ignore** the invitation, which requires an explicit action by User B.
- **Block** User A from sending more invitations.
- **Not answer**, that is do nothing with that request

Invitation to share Social Presence Information automatically implies the authorization of the requesting user, that is, when User A invites User B to share Social Presence Information, User A automatically authorizes User B to see their Social Presence Information.

If User A's MSISDN is associated with a contact in User B's address book, the name given to that contact shall be displayed within the invitation to share Social Presence Information.

Symmetric authorization shall be used. The publication of Social Presence Information shall be bidirectional.

User A shall not receive any notification whether User B has not answered, blocked or ignored their invitation to share Social Presence Information.

Once a Social Presence Relationship has been established, the user can stop that relationship via the following action:

Revoke the Social Presence Relationship.

3.7.1.4.2 Accept

If User B accepts User A's invitation to share Social Presence Information, User A will see User B's Social Presence Information, and User B will see User A's Social Presence Information.

If User A is not an existing contact in User B's address book, it shall be facilitated that User B stores the contact details of User A in their address book.

3.7.1.4.3 Ignore

If User B ignores User A's invitation to share Social Presence Information, neither User A nor User B shall be able to see each other's Social Presence Information.

Ignoring an invitation to share Social Presence Information shall not mean blocking the contact that has sent the invitation, i.e. it shall still be possible to receive more invitations from that contact.

If User B ignores User A's invitation to share Social Presence Information but later, User B decides to share their Social Presence Information with User A, then it is not necessary that a new authorization request is issued to User A. User B, by adding User A to their EAB

completes the symmetric authorization process. As a result, User A and User B will be seeing each other's Social Presence Information.

3.7.1.4.4 Block (refuse to receive any further invitation)

In order not to receive more invitations from a certain contact, the user shall be given the possibility to add that contact to a list of blocked contacts (blacklist).

The blocking mechanism shall be transparent to the blocked user, that is, if User B blocks User A, User A shall never be notified that he/she has been blocked by User B.

The possibility shall be given to remove a certain contact from the blacklist, i.e. User B shall be able to see in their EAB that User A has been blocked, and to remove them from the blacklist.

3.7.1.4.5 Not answer (pending invitation)

If User B does not answer User A's invitation to share Social Presence Information, the invitation shall be in a pending state, for which an action is expected by User B.

Pending invitations to share Social Presence Information with User A, for which an answer has not yet been provided, shall be accessible for User B, so that User B can choose to answer the invitation that is Accept, Ignore or Block.

Subsequent invitations (from User A to User B) replaces User A's initial invite, and function as a reminder for User B that a corresponding action (on their part) regarding the invitation to share SPI is required. That is, User B needs to choose an option

- Accept,
- Ignore, or
- Block.

3.7.1.4.6 Revoke

Once a Social Presence Relationship has been established, the possibility shall be given to stop the sharing of Social Presence Information with a certain contact, while at the same time removing your Social Presence Information from that contact's EAB.

If User A revokes the Social Presence Relationship with User B, both users shall not receive any further updates of their Social Presence Information, according to the symmetry principle.

When User A revokes the Social Presence Relationship with User B, User B shall no longer be displayed as a presence enriched contact.

User B's Social Presence Information shall not be shown to User A

Only User B's contact details (vCard) shall remain visible in User A's address book (for example name, MSISDN, e-mail, and so on.

If User A revokes the Social Presence Relationship with User B, User B shall no longer have access to User A's Social Presence Information.

Before actually performing the revoke, User A shall see a notification alert in the client informing him about consequences of this action. These are:

- User A's Social Presence Information will be removed from User B's EAB, so User B will notice the revoke after a certain period of time (for example several hours/days)
- It will be possible for User A and User B to invite each other again.

After a Social Presence Relationship has been revoked for a given period of time (for example several hours/days), both users can reinitiate the process of Social Presence

Authorization, that is User A shall be able to invite User B to share Social Presence Information, and vice versa.

It must be noted that User A may immediately re-invite User B to share Social Presence Information.

If User A deletes User B's vCard from their address book, all contact information is deleted from User A's address book. If a Social Presence Relationship between User A and User B exists at the moment of deleting the contact, this relationship shall be revoked.

3.7.1.4.7 Personalized Invitation

To improve RCS invitation experience with a personalized invitation and to ease identification of invitations coming from contacts not yet registered in the user's address book, a nickname feature is provided:

- If the terminal supports configuring a nickname, the user can choose a "nickname" with limited size (recommendation: 20 characters, this size can be set as a provisioning parameter). This nickname is provided in all future invitations to share presence, until it is changed. The maximum number of characters an invitee can view is 200 (this limitation is proposed to ensure interoperability for invitee, regardless of the number of characters implemented by the service provider).
- The invitee, if they do not have the inviter information in their address book, can now see both MSISDN and the nickname of the inviter.
- The nickname is stored permanently to be used for every invitation. Users have the ability to change it every time they send an invitation.
- The nickname does not replace the registered name of a contact already present in the recipient's address book.

Security: it is noted that through the use of the nickname, it is possible to "impersonate" someone. However, that "impersonation" is limited in scope since the inviting user remains identified by his MSISDN and the fact that the feature is only used for MSISDNs that are not already stored in the recipient's address book.

3.7.1.4.8 Geolocation authorization

Two users should be able to see where they are located and share this information with each other and they would keep the control over this information:

- No specific invitation process for location.
- When and if a user chooses (by opt-in) to update their location for the first time, by default, users do not share their location information with all their contacts authorized for social presence
- Users have the ability to manually choose contacts with whom they wish to share location information.
- Even if a user is not sharing location information with one of their authorized contacts, that does not prevent them from viewing that contact's location information

Once User A has accepted User B as an RCS authorized contact, User B will be able to see the geolocation information of User A (displayed with a text or a map, or both of them) and all updates of that information.

When a given RCS user (User A) is willing to share Social Presence with another user, User A shall be able to control in the invitation process for sharing Social Presence whether sharing of their location information with this other user is authorized or not.

3.7.1.4.9 VIP contacts

As the number of SPI enabled contacts increases in the user's address book, the amount of information that the user receives in the mobile phone will increase making it more difficult to differentiate useful information from noise. In addition, the RCS users will not want to share the same Social Presence Information with all their contacts.

The selection of certain contacts as Very Important Person (VIP) contacts will allow the end user to specify which contacts are the most important ones.

The user should be able to differentiate the contacts, which they share SPI with, between important and unimportant contact. The user shall receive real time notification of status changes from VIP contacts.

The user will be able to choose from the contacts to set them as VIP contacts. The user will then only receive real time notifications of the social presence information from the contacts set as VIP contacts (probably with a phone buzz or light, or via an idle screen widget and so on). The contacts that are not set as VIP contacts will still be updated in the EAB, but not in real time, therefore the user is made aware of the new social presence information when they browse the EAB.

3.7.1.5 Example Use Cases

3.7.1.5.1 Social Presence Information Use Cases

3.7.1.5.1.1 Invite Contacts to Share Social Presence

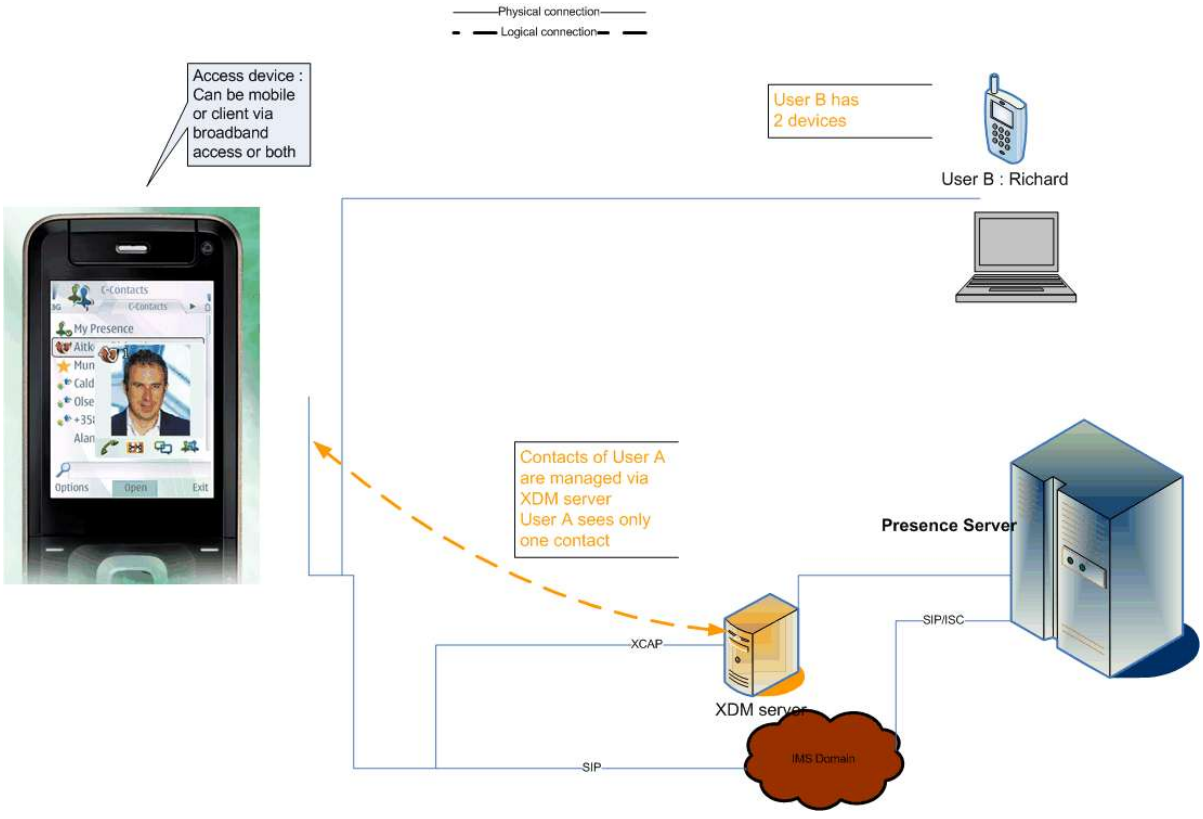


Figure 108: Invite Contacts to Share Social Presence

Authorization to share social presence is based on the symmetry principle. If sharing of social presence is accepted after invitation, both parties will see each other's presence attributes. If social presence sharing is terminated by one of both parties, both parties will end seeing each other's social presence attributes.

It is possible to share with an invitation for social presence a nickname if the invited party does not have the inviting party's phone number in the device.

3.7.1.5.1.2 Allow Contacts to obtain Location Information

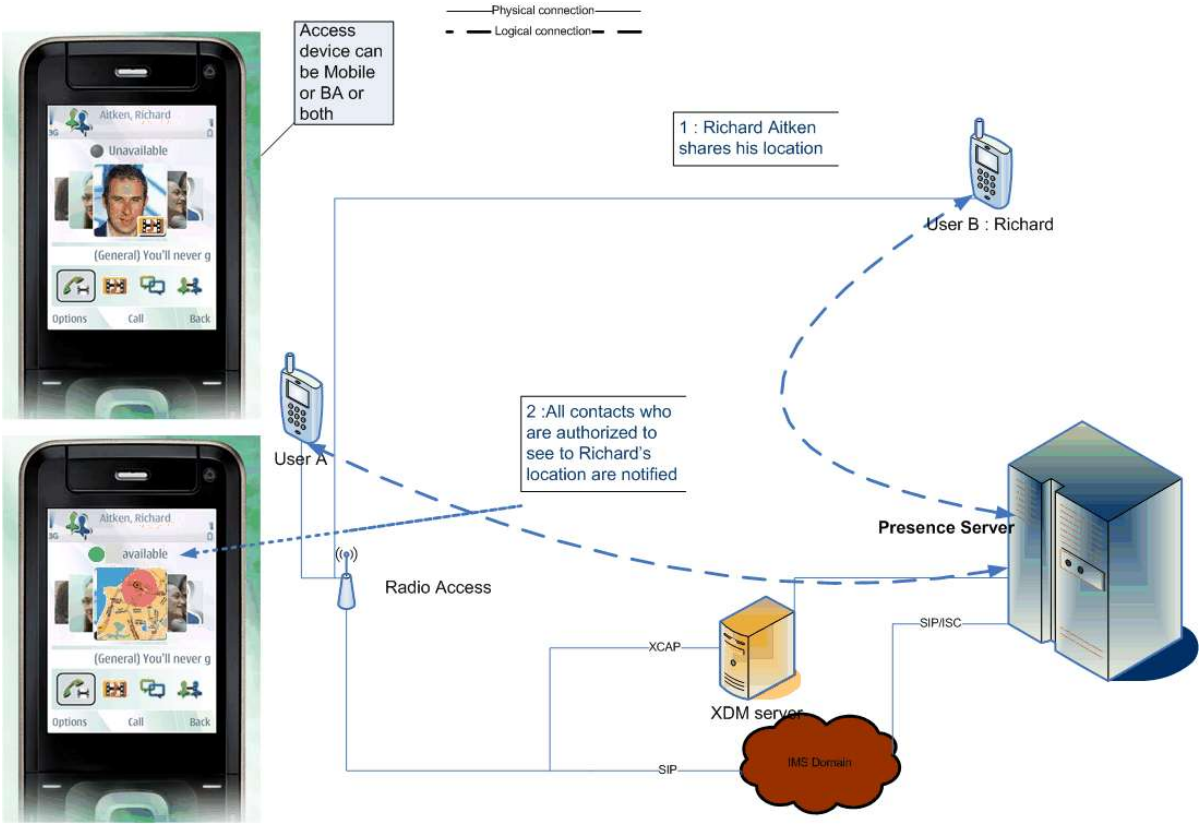


Figure 109: Share Location

This service allows users to show where they are through the RCS EAB and view where their friends are as free text and/or on a map.

NOTE: If the contact that is updated but not in the VIP group the information (Richard Aitken in the use case) in their VIP group may not be seen immediately. They will only see it when either their client polls for updates of the non-VIP contacts or when they request for an update of the non-VIP contacts themselves. The user may even miss the update altogether if there is another update before the status of the non-VIP contacts is retrieved.

3.7.1.5.1.3 Availability

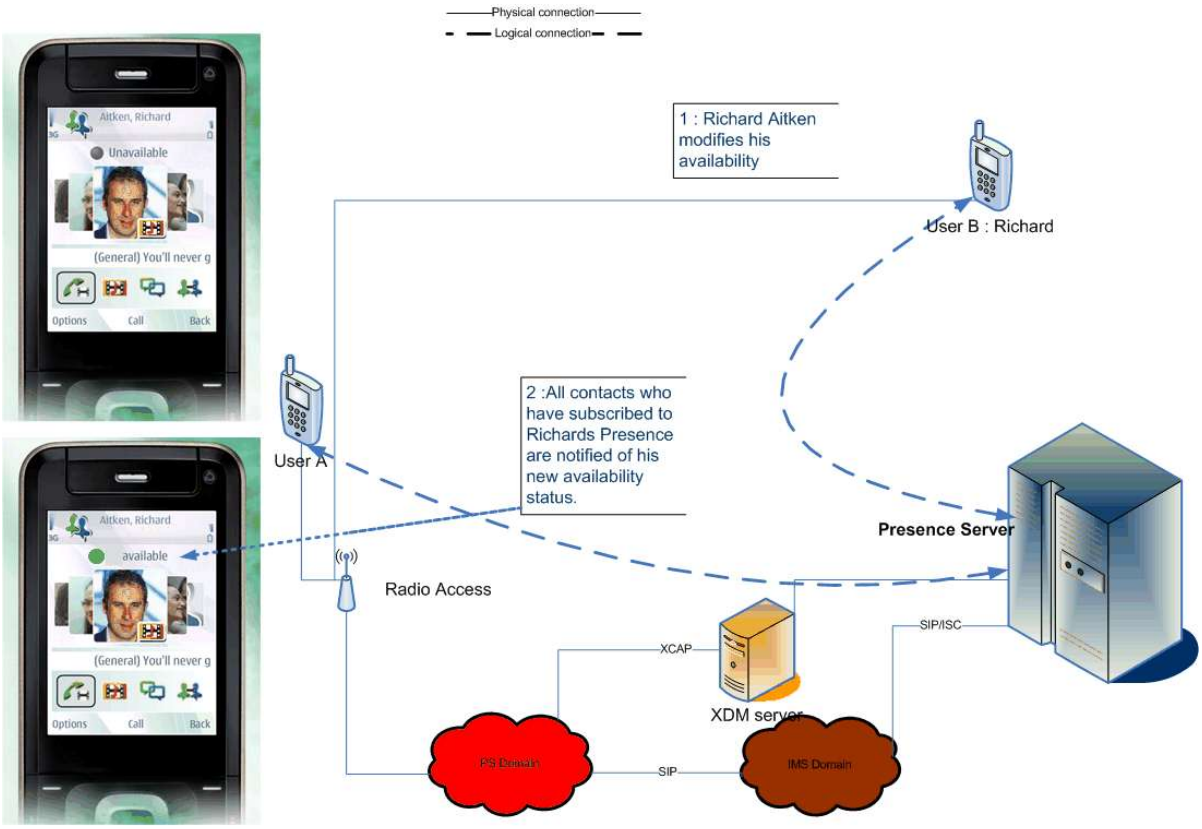


Figure 110: Availability

NOTE: If the contact that is updated is not part of the VIP group of the user the updated SPI (Richard Aitken's in the use case) may not be seen immediately. They will only see it when either their client polls for updates of the non-VIP contacts or when they request for an update of the non-VIP contacts themselves. The user may even miss the update altogether if there is another update of the availability status before the status of the non-VIP contacts is retrieved.

3.7.1.5.1.4 Free Text

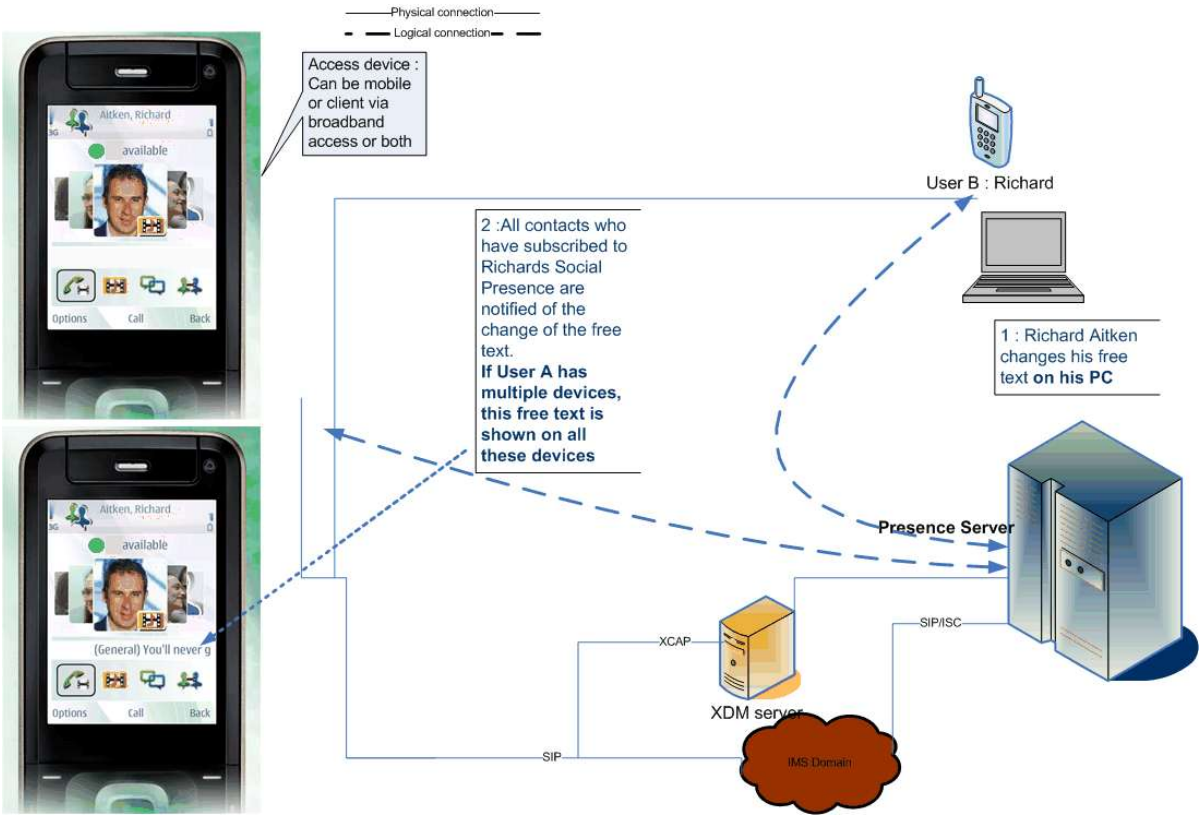


Figure 111: Free Text

NOTE: If the contact that is updated is not part of the VIP group of the user, the updated SPI (Richard Aitken's in the use case) may not be seen immediately. They will only see it when either their client polls for updates of the non-VIP contacts or when they request for an update of the non-VIP contacts themselves. The user may even miss the update altogether if there is another update before the status of the non-VIP contacts is retrieved.

3.7.1.5.1.5 Portrait Icon Exchange

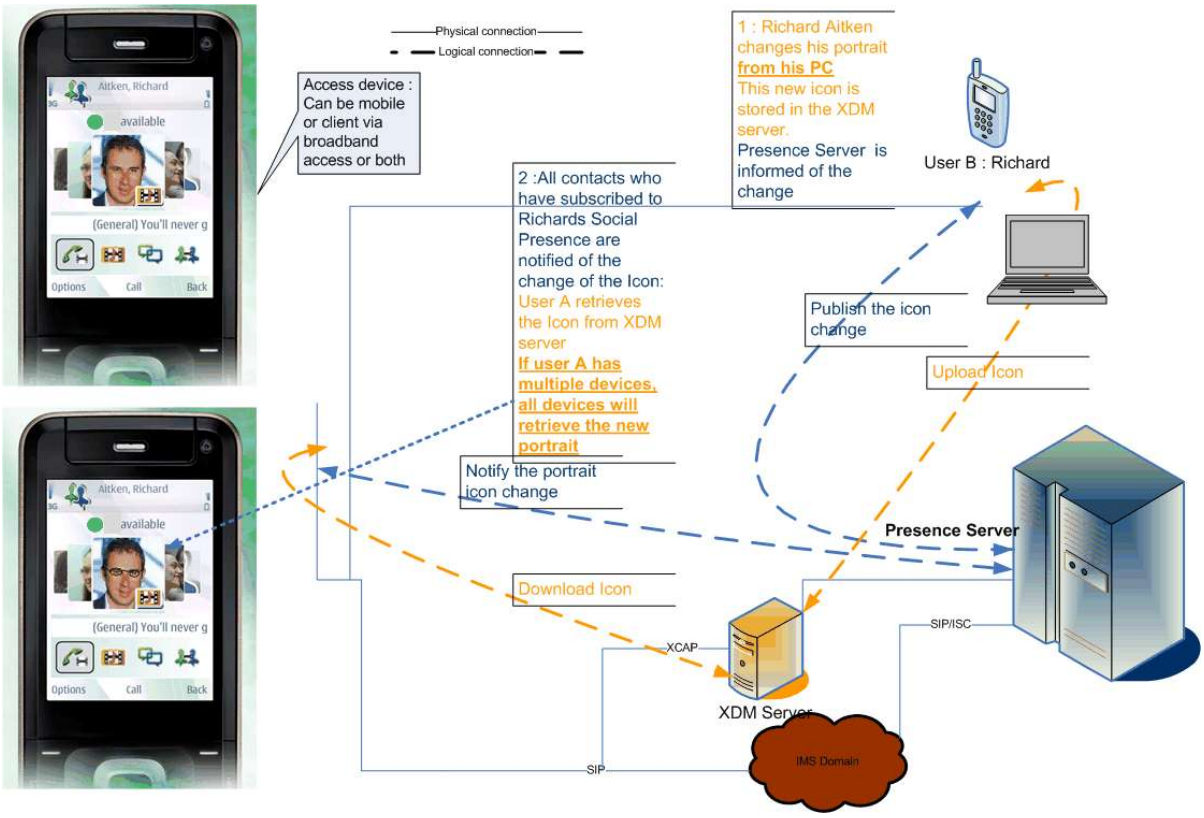


Figure 112: Portrait Icon Exchange

NOTE: If the contact that is updated is not part of the VIP group of the user the updated SPI (Richard Aitken's in the use case) may not be seen immediately. They will only see it when either their client polls for updates of the non-VIP contacts or when they request for an update of the non-VIP contacts themselves. The user may even miss the update altogether if there is another update before the status of the non-VIP contacts is retrieved.

3.7.1.5.1.6 Who Can I Invite?

New user wants to invite their friends to share social presence.

- User A goes to their RCS enhanced address book
- User A traverses through the list of contacts and sees that User B is also an RCS user that supports the SPI service based on the capability discovery mechanism defined in section 2.6
- User A decides to send an invitation to share Social Presence Information to User B.

3.7.1.5.2 Personalized Invitation with a Nickname

3.7.1.5.2.1 User A invites User B and fills out their Nickname. User A is present in User B's address book

- When User B receives the invitation, it is the contact name entered in User A's v-card that is used, not the nickname.
  - For example, User B can read "<User A v-card name> <MSISDN> wants to share presence information with you."

3.7.1.5.2.2 User A invites User B and fills out their Nickname. User B has not created a contact card for User A in their address book

- When User B receives the invitation, the nickname is used to present the invitation to User B
  - For example, User B can read "<User A nickname> <MSISDN> wants to share presence information with you."
- If User B accepts the invitation, a contact card is created. User A's nickname can be used to reference the contact card in User B's address book.

3.7.1.5.3 Geolocation

3.7.1.5.3.1 Manual Free Text

- User A set his location manually (for example, I am in Paris)
- User B sees that User A is in Paris.

3.7.1.5.3.2 Manual Position on a Map

- User A decides to update their current location. User A drags and drops a pin on a map and then confirms the position. Even though User A is located in Paris, France, they select New York as a location on the map.
- User B receives a notification.
- User B sees that User A is in New York.

3.7.1.5.3.3 Semi-Automatic Filling

User A decides to edit their current location status. User A selects the location update button, and their location is automatically filled in the dedicated field used to enter their location.

3.7.1.5.3.4 Fully Automatic Opt-In Mode

User A decides that they want their authorized contacts to be informed regarding their position on a regular basis (period to be defined), they click on the "authorize my contacts to view my location" button (opt in). If they decide to end this broadcast they always have the ability to opt out through the same button.

In all cases, User B (authorized contact in User A's address book) is notified as he would be notified of other presence information, such as status text.

3.7.1.5.3.5 Blocking an Authorized Contact from Viewing Location

- User A and B are authorized RCS contacts who have updated their location information
- User A decides to hide their location from User B, while still sharing it with his other authorized contacts
- User A goes to his location settings currently set to "Share my location with all my authorized contacts" to "Prevent some authorized contacts from viewing my location"
- User A adds User B in the list of contacts blocked from viewing their location
- User B does not see User A's location information anymore
- User A still sees User B's location

3.7.1.5.4 VIP Contacts

3.7.1.5.4.1 User sets a contact as a VIP

User A is an RCS user.

User B is an RCS user.

User A and User B had established a Social Presence Information sharing relationship.

Call Flow:

- User A sets User B Contact as a VIP contact in their address book.
- User B changes their Social Presence Information.
- User A receives an active notification (phone buzz or light, idle screen widget) about the change.

3.7.1.5.4.2 User sets a contact as a non-VIP

User A is an RCS user.

User B is an RCS user.

User A and User B had established a Social Presence Information sharing relationship.

User A had previously set User B as a VIP contact.

Call Flow:

- User A sets User B Contact as a non-VIP contact in their address book.
- User B changes their Social Presence Information.
- User A does not receive any active notification about the change but if they access later their EAB and browse to the User B contact, the EAB will display the changed information.

3.7.2 Interaction with other RCS features

Social Presence information in the device is linked with the local address book available in the device:

The social information elements of a contact in an RCS device are, from user interface point of view, associated (as an extension of other address book contact information) with the contact entry of the address book.

This correlation is local:

- Local contact information may be synchronised with a Network address book
- Extended presence information is obtained through the Network Presence enabler

3.7.3 High Level Requirements

3-7-1 An RCS user with broadband access shall be able to access the Enhanced Address Book, supporting all the social presence features.

3-7-2 A broadband access client should support Social Presence Authorization.

3-7-3 The presentity shall be able to edit the Social Presence Information from any of the devices he/she has and shall see the changes from every device he/she has

3-7-4 Social Presence Information shall be handled in such a way that the latest update is presented to the watching user's client

3-7-5 The invitation to share Social Presence Information shall be shown in all of the presentity’s devices

3-7-6 The presentity shall be able to authorize watchers from any of the devices they have

3-7-7 If a certain setting may limit the user experience provided to the end user, this information should be clearly shown in the user interface. In addition this allows the user to be aware of this limit while interacting with the service (for example, maximum number of characters to be included in the free text of the Social Presence Information, or maximum size of a file to be transferred).

3-7-8 The User shall be able to share location information as social presence information with his/her authorized contacts

3-7-9 The User shall be able to define a list of contacts blocked from viewing his/her location information, within his list of authorized contacts for presence

3-7-10 The User shall be able to specify their location through manual or automatic modes, as free text or as coordinates on a map

3-7-11 The User shall be able to de-activate automatic updates or delete their location information at any time, to protect their privacy

3-7-12 The User shall be able to share location information even if he/she is using a non-GPS device

3-7-13 The Service Provider shall be able to limit the frequency of automatic updates to avoid network overload

3-7-14 The RCS user shall be able to set an expiration date for location information

3-7-15 The User shall be able to define a nickname transmitted to his contacts when sending invitations, in addition to the MSISDN

3-7-16 The User shall be able to change that nickname at any time, especially before sending invitations

3-7-17 The Service Provide shall be able to specify the maximum length of the nickname

3-7-18 The Nickname shall never automatically replace the existing registered name of a contact in the invitation recipient’s phonebook

3-7-19 The User shall be able to specify a text label displayed in lieu of the personal URL

3-7-20 The User shall be able to change the URL label at any time

3-7-21 The Service Provider shall be able to specify the maximum size of the URL label

3-7-22 An RCS user shall be able to set a contact as a VIP contact.

3-7-23 An RCS user shall be able to unset a contact as VIP.

3-7-24 When a VIP contact updates his Social Presence Information the user shall get a real time notification of the change and it shall be displayed on the RCS client (phone buzz or light indication, idle screen widget).

3-7-25 When a non-VIP contact updates their Social Presence Information, the user shall not be notified in real time about the changed status. The RCS client shall keep that information up to date (but not in real time) so the contact information is updated when the user browses the EAB.

3-7-26 The update mechanism for updating non VIP contacts shall be a periodic polling mechanism from the RCS client resulting in an aggregated notification from the network. The update period shall be configured by the RCS Service Provider by parameter.

3-7-27 In addition, an RCS user shall be able to manually request an update of all the non-VIP contacts.

3.7.4 Technical Realization

3.7.4.1 Network architecture of Presence enabler in RCS

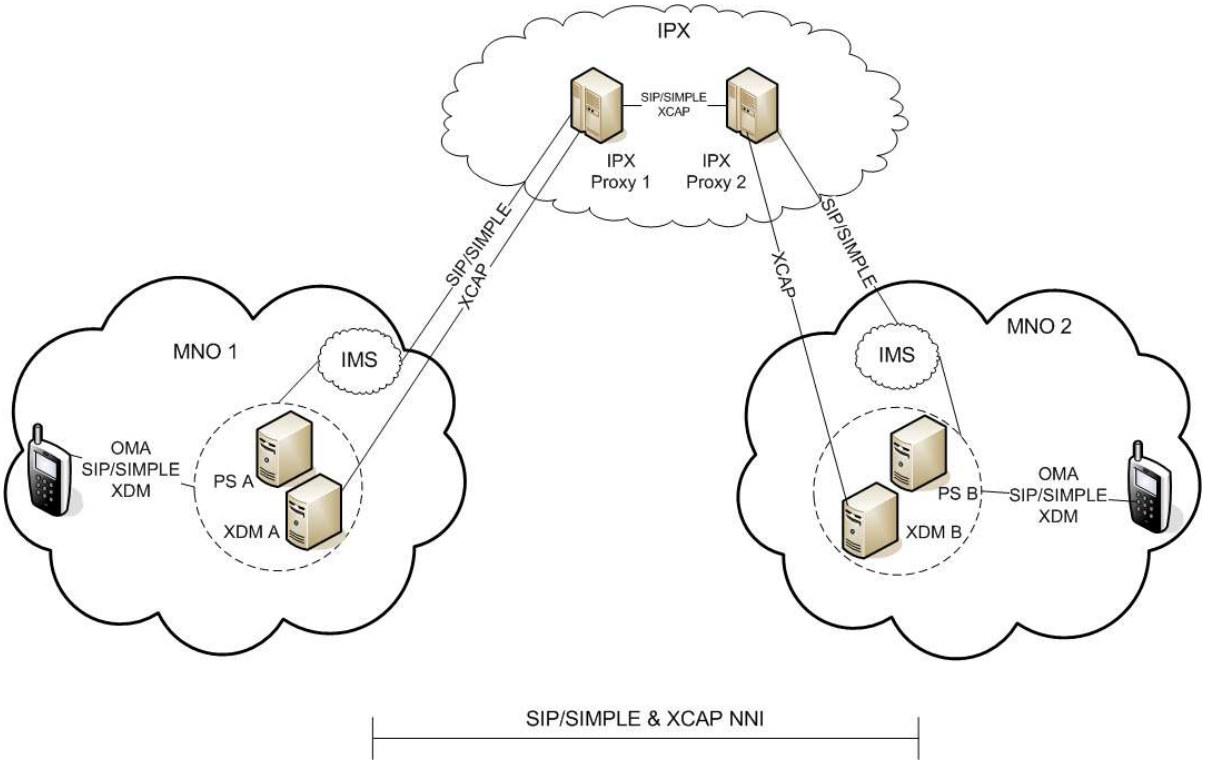


Figure 113: Overall Architecture of Presence as a part of RCS

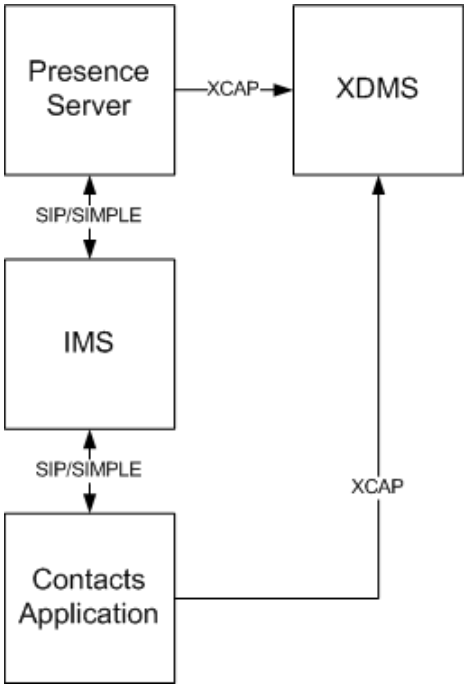


Figure 114: RCS Presence Architecture

Presence and capability architecture in RCS is based on [Presence].

Users share their Social Presence Information (“Presence Enhanced Address Book”)

- Implemented using the Presence SIMPLE protocol

Users share their communication capability information (“Capability Enhanced Address Book”)

- Can be implemented using the Presence SIMPLE protocol (see section 2.6.1.2)

According to [PRD-IR.65], the interworking connection should be carried out via IMS core systems. There is therefore no requirement to interface Presence Servers directly.

Optimization of Presence & XDM enabler according to work in OMA PAG working group has to be taken into account as a very important design principle. It is also important to notice potential issues such as battery drain in the terminal caused by the general always-on functionality and the number of Presence & capability updates.

Generally, the Shared XDMS (XDM server) as defined in [XDM1.1_AD] shall be used for storing all presence-related lists, for example, the list of subscribed contacts (“buddy” list) and the presence authorization lists. In this way, the RCS client only needs to operate on lists in Shared XDMS, and initially set the documents in RLS (Resource List Server) XDMS and Presence XDMS.

3.7.4.2 Presence Data Model

3.7.4.2.1 Overview

Implementation guidelines for the size/length of Presence information elements given in [PRESENCEIG] should be followed.

The following sections illustrate the details of the *Person* and *Device* parts of the Presence Data Model. The Service part of the model has been described in section 2.6.1.2.5.

3.7.4.2.2 Person

Attribute	Specification	Comment
Person: <presence> -> <person>	[Presence2.0_DDS]	According to the presence schema defined in the [Presence], person related information is modelled with the person element. Each client only publishes one person element.
Willingness: <person> -> <overriding-willingness> -> <basic>	[Presence2.0_DDS]	The presentity terminal publishes this attribute in which it wants to indicate its willingness to communicate: “Open” = Willing “Closed” = Not Willing Attribute not present = Unknown
Icon: <person> -> <status-icon>	[Presence2.0_DDS]	It’s used as dynamic avatar. If the element is not present the client may choose to display icon stored in the address book. The picture shall not be included directly in the presence requests, but a HTTP URL shall be used. Presence Content XDMS procedures as specified in OMA Presence 2.0 and XDM 2.0 is used for uploading, publishing and retrieving the icon For further details see section 3.7.4.2.2.2

Favourite Link : <person> -> <link>	[Presence2.1_DDS]	The <link> element provides a URI pointing to general information about the tuple or person, typically a web home page. This is information is complemented with a “label” attribute set to a value provided by the served RCS presentity and a priority attribute which is intended to cope with situations in which there are multiple <link> elements. In RCS only one such <link> element will be included in the presence document though. The priority attribute will therefore always be set to 0.8.
Descriptive Location Text <person> -> <place-type> -> <other>	[Presence2.0_DDS]	The presentity may provide a descriptive text describing his location See section 3.7.4.2.2.3 for more information on the handling of the expiry of this information NOTE: Support for the enumerated values defined in [RFC4589] is thus out-of-scope for RCS. It is out of scope of RCS how a client will handle these enumerated values when received nevertheless.
Time Zone <person> -> <time-offset>	[Presence2.0_DDS]	The presentity may use this element to provide information on his current time zone See section 3.7.4.2.2.3 for more information on the handling of the expiry of this information
Geographical Information <person> -> <geopriv> -> <location-info> -> <usage-rules>	[Presence2.0_DDS]	This element can be used to provide geographical location information on the presentity. The accuracy of which can be controlled by the user. See section 3.7.4.2.2.3 for more details on its encoding and on the handling of the expiry of this information
Note: <person> -> <note>	[RFC4479]	The presentity may write a piece of free text and/or to add emoticons to be shown to watchers in their contacts books The list of emoticons in RCS can be found in [RCS5-SIMPLEIM-ENDORS]
Timestamp: <person> -> <timestamp>	[RFC4479]	Timestamp when the presence information was published.

Table 69: Presence data model attributes

Note1: “Willingness” is sometimes indicated in a client as “Availability”. However since it is managed by the user themselves and does not imply that communication is not possible, within OMA specifications this is considered as willingness. Availability indicates that on a technical level communication will be possible. Service Availability and Willingness are study items for later releases.

Note 2: the priority of 0.8 for the link was included to allow including links with higher priority in some future RCS release.

3.7.4.2.2.1 Willingness

A Service Provider provisioning parameter (AVAILABILITY AUTHORISATION as described in section A.1.1.2) is provided indicating whether or not the use of willingness is enabled by the service provider. If it is disabled, no OMA <*overriding-willingness*> element is included in the presence document. If the willingness is enabled, the RCS client will include in the presence document an OMA <*overriding-willingness*> element as specified in [Presence2.0_DDS] with the <*basic*> sub-element set to "closed" when the user has indicated that he's not willing to communicate. Otherwise if willingness is enabled, the published presence document will indicate a value of "open" for the <*basic*> sub-element of <*overriding-willingness*>.

3.7.4.2.2.2 Icon

The icon shall have following characteristics:

Document Name	rcs_status_icon
Icon aspect ratio (width:height)	3:4 or 4:3
Icon maximum dimensions	240x320
Icon minimum dimensions	60x80
Icon file type	gif (Graphics Interchange Format, both static and animated), jpeg (Joint Photographic Experts Group) or png (Portable Network Graphics) as defined in [Presence_Content]
Document maximum size	200 kilobytes (KB)

Table 70: Characteristics of the icon

Note 1: Fixing the icon document name will ensure that for RCS usage, a single icon is stored in the network and no unnecessary resources are required for the storage of multiple icons. Without this, the situation could occur that multiple icons are stored without possibility to manage them after a switch to a new client. Furthermore the fixing of the icon name will allow clients that are aware of the SIP URI of their contact to build the URI needed for the retrieval of the icon even if the contact is offline.

Note 2: 200KB is not a mandatory size. It is only defined as a maximum and smaller sizes are acceptable

The other parameters are fixed to allow the client implementations to know what to expect.

3.7.4.2.2.3 Location Information

RCS clients shall not include a "from" attribute in the <place-type> and <time-offset> elements. RCS clients shall ignore it when received. RCS clients shall provide an "until" attribute in those elements and set it as specified in section 3.7.4.3.2.4.3.

RCS clients shall not include the optional description attribute in the <time-offset> element as this overlaps with the Location Type. RCS clients shall ignore it when received.

The geographical information will be provided as geographic coordinates. As specified for the "Geographical Location" building block in [Presence2.0_DDS], encoding will use the <*geopriv*>→<*location-info*> and <*geopriv*>→<*usage-rules*> elements.

The mandatory <*usage-rules*> element shall contain only a "*retention-expiry*" element as RCS clients will request the watchers to follow the default handling for the other rules. The RCS client shall set the "retention-expiry" as specified in section 3.7.4.3.2.4.3.

The <*location-info*> published by an RCS presence source will contain geographical information using the GML (Geography Markup Language) 3.1.1 Feature Schema (see [GML3.1.1]) which is the mandatory format to be used in the <*location-info*> element. The

civic location format shall not be used by RCS presence sources and location information encoded in that way will be ignored by RCS clients when received.

RCS presence sources will within the <*location-info*> element represent an exact position by providing a GML <*point*> element and an inaccurate position as a <*circle*> element, both referring to the EPSG::4326 spatial reference schema as described in [RFC5491]. The coordinates of either the centre of this circle or the exact position will be represented with a single GML <*pos*> element with the actual coordinates as value. The radius of the circle will be represented in meters, which will be indicated by setting the unit of measure attribute of the radius element to the value of EPSG::9001 as described in [RFC5491]. An RCS client shall ignore any other type of data provided in the <*location-info*> element.

The European Petroleum Survey Group (EPSG) format requires that the coordinate representation is defined by the coordinate supplier. RCS presence sources will always provide the coordinates in WGS 84 (latitude, longitude) decimal notion as described in [RFC5491], providing the latitude and longitude as “double”-encoded decimal numbers (as specified in [GML3.1.1]) representing the degrees, separated by a space starting with the latitude. Negative values represent Southern and Western hemisphere respectively.

3.7.4.2.3 Service

See section 2.6.1.2.5.

3.7.4.2.4 Device

The Device part of presence is out of scope for RCS.

3.7.4.2.5 Example Document

The above leads to following example document:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<presence xmlns=”urn:ietf:params:xml:ns:pidf”
            xmlns:op=”urn:oma:xml:prs:pidf:oma-pres”
            xmlns:opd=”urn:oma:xml:pde:pidf:ext”
            xmlns:opd11=”urn:oma:xml:pde:pidf:ext:1.1”
            xmlns:pdm=”urn:ietf:params:xml:ns:pidf:data-model”
            xmlns:rpid=”urn:ietf:params:xml:ns:pidf:rpid”
            xmlns:gp=”urn:ietf:params:xml:ns:pidf:geopriv10”
            xmlns:caps="urn:ietf:params:xml:ns:pidf:caps"
            xmlns:gml="http://www.opengis.net/gml"xmlns:gs="http://www.opengis.net/pidflo/1.0"
            entity=”tel:+1234578901”>

  <tuple id="a2">
     <status><basic>open</basic></status>
     <op:service-description>
        <op:service-id>org.3gpp.urn:urn-7:3gpp-service.ims.icsi.mmtel</op:service-id>
        <op:version>1.0</op:version>
     </op:service-description>
     <caps:servcaps>
        <caps:audio>true</caps:audio>
        <caps:duplex>
           <caps:supported>
              <caps:full/>
           </caps:supported>
        </caps:duplex>
     </caps:servcaps>
     <contact>tel:+1234578901</contact>
  </tuple>
  <tuple id=”a1”>
     <status><basic>open</basic></status>
```

```
    <op:service-description>
        <op:service-id>org.3gpp.cs-videotelephony</op:service-id>
        <op:version>1.0</op:version>
    </op:service-description>
    <contact>tel:+1234578901</contact>
</tuple>

<tuple id=”a12”>
    <status><basic>open</basic></status>
    <op:service-description>
        <op:service-id>org.gsma.videoshare</op:service-id>
        <op:version>1.0</op:version>
    </op :service-description>
    <contact>tel:+1234578901</contact>
</tuple>
<tuple id=”a123”>
    <status><basic>open</basic></status>
    <op:service-description>
        <op:service-id>org.gsma.videoshare</op:service-id>
        <op:version>2.0</op:version>
    </op :service-description>
    <contact>tel:+1234578901</contact>
</tuple>
<tuple id=”a132”>
    <status><basic>open</basic></status>
    <op:service-description>
        <op:service-id>org.openmobilealliance:IM-Session</op:service-id>
        <op:version>1.0</op:version>
    </op :service-description>
    <contact>tel:+1234578901</contact>
</tuple>

<pdm:person id=”a1233”>
    <op:overriding-willingness>
        <op:basic>open</op:basic>
    </op:overriding-willingness>
    <rpid:status-icon opd:etag=”26362”>http://xcap.gsma.org/xcap-ap/service/org.openmobilealliance.pres-
    content/users/sip:1234578901@gsma.org/oma_status-icon/rcs_status_icon</rpid:status-icon>
    <opd11:link opd11:label=”my blog” opd11:priority=”0.8”>
        http://example.com/~alice
    </opd11:link>
    <rpid:place-type opd:until=”2009-11-28T21:00:00Z”>
        <rpid:other>Herentals, Belgium</rpid:other>
    </rpid:place-type>
    <rpid:time-offset opd:until=”2009-11-28T21:00:00Z”>+120</rpid:time-offset>
    <gp:geopriv>
        <gp:location-info>
            <gs:Circle srsName="urn:ogc:def:crs:EPSG::4326">
              <gml:pos>51.1644 4.7880</gml:pos>
              <gs:radius uom="urn:ogc:def:uom:EPSG::9001">10</gs:radius>
            </gs:Circle>
        </gp:location-info>
        <gp:usage-rules>
            <gp:retention-expiry>2009-11-28T21:00:00Z</gp:retention-expiry>
        </gp:usage-rules>
    </gp:geopriv>
    <pdm:note>I’ll be PAG</pdm:note>
```

```
  </pdm:person>
</presence>
```

Table 71: Example Presence Document

3.7.4.3 Presentity Side Handling

3.7.4.3.1 Publication Methods

3.7.4.3.1.1 Overview

An RCS client publishes its presence information using two different methods:

1. SIP PUBLISH requests
2. Permanent Presence State Publication (that is, a permanent document maintained through XCAP)

The method to be used depends on the information to be published:

SIP PUBLISH requests are used for following data:

- Service Capabilities

Permanent Presence State publication applies to the following attributes of Social Presence Information:

- Portrait icon
- Free text
- Favourite link
- Willingness (that is the overriding-willingness element)
- Location Information

3.7.4.3.1.2 Permanent Presence State Publication

The RCS Client shall support Permanent Presence State publication by manipulating the Permanent Presence State via an XDMC using the permanent presence state application as defined in [Presence2.0_TS]. An RCS client shall update the permanent presence state document in such a way that elements in the document that are not changed or are even unknown to the RCS client (for example, because they were included by a client supporting a future RCS release), are not altered. To avoid inconsistencies between attributes and the actual element value, unknown attributes of changed elements shall be removed from the updated document.

This can be achieved both through a direct, conditional update of only the changed element itself or through a retrieval of the complete document followed by a client local update of the changed elements. This update should then be used in a conditional replace request for the entire permanent presence state document. The choice between both methods is left to client implementation and could depend on the amount of updated elements. In both cases, whenever the document is modified any expired information will be removed (for example Location Information with an “until” attribute indicating a time in the past).

The RCS Presence Server shall use the Permanent Presence State as input for Presence Information processing. RCS Presence Server should subscribe/fetch the permanent presence state document from Presence XDM when applying the composition policy.

3.7.4.3.2 Presence Information Handling

3.7.4.3.2.1 Willingness

When the Service Provider provisioning parameter indicates that willingness is enabled, at the presentity side, the RCS client will always include an <*overriding-willingness*> element in the permanent presence document. This element will have a <*basic*> sub-element set to

either "*open*" or "*closed*" depending on what was indicated by the user as his current status. If willingness is disabled through the provisioning parameter, no "<*overriding-willingness*>" element will be included in the permanent presence document.

3.7.4.3.2.2 Status Icon

The status icon shall be stored, updated, deleted and retrieved according to the OMA Presence and XDM 2.0 procedures. For the storage itself, the Presence Content XDMS as defined in [Presence_Content] shall be used including the application usage and document type that it introduces. RCS will only make use of the Presence Content XDMS for the storage of the status icon. Therefore the usage as defined in section 5.1.12.1 of [Presence_Content] is the only one that is applicable including all its associated restrictions. After storing, updating or deleting the icon, the presentity's client should publish an updated presence document including the *etag* attribute in the <*status-icon*> element as described in [Presence2.0_DDS] in sections 7.11.1.3 and 7.20.

3.7.4.3.2.3 Link

The RCS client will limit the length of the label to the maximum length that is provided through a Service Provider provisioning setting.

3.7.4.3.2.4 Location

3.7.4.3.2.4.1 Ending Location Information Sharing

When the user indicates that they do not want to share their location information with the contacts allowed to see their information anymore, the client can fulfil this request by removing the location information from the Permanent Presence State document.

3.7.4.3.2.4.2 Obtaining Location Information

See section 3.10.4.4.

3.7.4.3.2.4.3 Managing Location Information

An RCS presence source is not required to include all location elements specified in section 3.7.4.2.2.3 in the permanent presence state document (that is, all elements are optional to be provided).

The length of the descriptive text that the RCS client includes in the Permanent Presence State document shall not be longer than the maximum that was provided as a Service Provider provisioning setting.

The maximum time a location update remains available to watchers is controlled by a Service Provider provisioning setting. RCS presence sources will set the "*until*" attribute and the "*retention-expiry*" element (see section 3.7.4.2.2.3) in accordance to this provisioning setting (that is, set it to the current time increased with the value of the setting). Furthermore RCS presence sources shall remove expired location information from the published presence document and from any locally cached copy of that document whenever they update other elements in the document.

Clients offering the user the choice to provide an inaccurate position to their contacts (for example, city level or even country level) can do so by providing a *CircleByCenterPoint* element instead of an exact position using coordinates and text reflecting this inaccuracy (for example, the city centre instead of the exact street). Whether the client does this and how it determines the position of the centre, the radius and the text value (that is, the <*place-type*> element) that will be shared, is considered to be client implementation and thus out-of-scope for RCS.

As an option to the user, clients may also offer the possibility to regularly update their position without user intervention. Whether or not this is done is again considered to be a

client implementation issue and thus out-of-scope for RCS. Since such an implementation could result in a high load on the network and the clients of the contacts with whom location is shared, some Service Provider control is required. This will be realized through a Service Provider provisioning setting controlling the minimum duration between location updates. An RCS client shall ensure that the time between two consecutive location updates is larger than this provisioned minimum.

NOTE: Even though a maximum update frequency could be derived from the provided minimum duration setting, it has been an explicit choice not to provision a frequency, as no updates would be necessary if the device has not moved. Again the decision on when an update is needed is left to the client implementation and thus out-of-scope for RCS provided the client complies with the provisioned minimum interval between updates.

3.7.4.3.2.5 Nickname

The *application/watcherinfo+xml* body in the watcher information notification may contain a display name for the watcher in the display-name attribute as specified in [RFC3858]. In this case, if the telephone number that is derived from the (SIP or tel) URI that is provided for that watcher is not found in the phone book of the client, the RCS client will include the display name in notifications shown to the user. At the same time it will always include the watcher's telephone number to minimize the risk of false identifications.

If no display name is received (for example because the subscription is initiated from an RCS Release 2 network), the client shall only present the E.164 number to the user.

If the watcher's telephone number is found in the phone book, behaviour shall be as specified in section 2.5 (that is, the received display name shall not be used, but rather the information that is part of the phone book).

An RCS client shall be able to deal with display names up until a maximum length of 200 characters.

3.7.4.3.3 Multidevice Handling

If one of the user's clients changes the (shared) permanent presence state document, the other clients of the user will receive the update as part of a presence notification which will contain information about their own presentity. Such an update will be received immediately when the client is online at the time of the changes. If this is not the case, the client will receive the update when it comes online. Clients shall take the updated social presence information into account and update the presence information that they store locally in the client accordingly. To get the notifications that are necessary to provide this behaviour, the client shall include the own identity in the "*rcs*" list which is part of the Shared XDMS's "*resource-lists*" document (see section 2.14.1).

When a user decides that they do not want to receive a certain service on one of their secondary clients (see section 2.9.1.4), the given secondary client will not indicate the capability for that service in the services section of the presence document if such a capability is defined for the service (see section 2.6.1.2.5).

3.7.4.4 Watcher Side Handling

When presence information of a presentity is requested by a watcher a SUBSCRIBE request is initiated (event package '*presence*') according to [Presence]. The watcher should be able to use the tel URI to identify the presentity, see section 2.5.

The support of RLS is mandatory for the clients and servers. Client shall conform to section 5.2.2.1 of the technical specification of [PRESENCE] and in addition to section 5.7.1 and 5.8

in [PRESENCEIG], section 5.1 in [XDMIG] and section 5.1.6 in [RLSXDM]. The XML documents shall follow the templates following later in this section.

3.7.4.4.1 Caching Presence Information

The caching of presence information is a client procedure.

The RCS client must be able to locally store the most up-to-date presence information (that have been received through notifications) of all of the user's contacts. This locally stored information must be handled as a persistent cache (that is the data shall not be erased when the terminal is switched-off).

3.7.4.4.2 Presence Information Handling

3.7.4.4.2.1 General Processing Rules to Facilitate Forwards Compatibility

To maintain enough flexibility and not to impose potentially sub-optimal technical choices on future RCS releases, the presence parsing for social presence information in an RCS client should be sufficiently robust. Therefore the following guidelines should be taken into account in RCS presence parsing:

- Unknown or unsupported elements could be present in the document. In that case they should be ignored.
- When using RLS subscriptions, information could be contained on presentities that were not known to be part of the presence list (for example because the list was updated by another client or application). If the unexpected presentity is a known contact, the client should treat this contact as being presence enabled (see section 3.7.4.4.4) and try to retrieve an updated presence list from the network (see section 2.14.2).
- The Watcher shall follow the procedures defined in section 6.2 "Default Watcher Processing" of [Presence2.0_DDS].

3.7.4.4.2.2 Willingness

When the service provider provisioning parameter indicates that willingness is disabled, on reception of a NOTIFY request, the watcher RCS client will ignore any "<*overriding-willingness*>" in the received presence document(s). If willingness is enabled the client will interpret any "<*overriding-willingness*>" element included/not included in the received presence document(s) as specified in section 3.7.4.2.2.

3.7.4.4.2.3 Status Icon

The link to the status icon that is received in the presence document of the contact will be processed as described in [Presence2.0_TS] section 5.2.5.3. When the *etag* attribute of the *status-icon* element does not match that of the cached icon, the client will download the updated icon. To do that it will handle the link that it received in the presence document as defined in [XDM2.0_Core] section 6.1.1.1 and more specifically the third paragraph: it will replace the XCAP root part of the link with the own XCAP root of the watcher. After downloading the icon, the RCS client shall cache it along with the *etag* to be able to process future notifies on the status of the contact as defined in [Presence2.0_TS] section 5.2.5.3.

3.7.4.4.2.4 Link

If an RCS client receives a document containing multiple <*link*> elements, then it shall only consider the one with the highest priority and use that as the value of the <*link*> element in the processing.

An RCS watcher shall be able to deal with labels with a length of maximum 200 characters.

3.7.4.4.2.5 Location Information

It is considered to be a client implementation decision how received location information from a contact will be handled (for example, display only the text, use an individual map for each contact and so on. This is thus considered to be out of scope for RCS. Clients should at least provide a means to display any descriptive text (that is, the content of the <*place-type*> element) that they might receive.

An RCS client should take into account that a received presence document might not contain location information (for example, because the presence source does not provide it or privacy was enabled).

An RCS client shall be able to deal with *place-type* information with a length of maximum 200 characters.

An RCS client shall not display to the user information contained in location elements for which the "*until*" attribute (for the <*time-offset*> and <*place-type*> elements) or the <*retention-expiry*> element (for the geolocation information) indicate a time in the past. Furthermore it shall not cache the expired information locally any longer.

3.7.4.4.3 Nickname Handling

If the user has provided a nickname, an RCS client shall include it as the display name as part of the identity information provided in the *P-Preferred-Identity* and *From* header field of the SIP SUBSCRIBE request used when subscribing to the user's Resource List Server (RLS) document. The RCS client shall ensure that the length of the used display name is not larger than the maximum size that was provisioned by the Service Provider.

3.7.4.4.4 Multidevice Handling

For the most part the watcher functionality on the different clients of the same user can function independently of each other. Only with the authorization there might be some interaction as this may trigger unexpected notifications (see section 3.7.4.5.7). An RCS client of this release will provide compatibility with clients of future RCS releases acting as one or more of the other devices of the user. To achieve this it will display the presence information provided in a presence notification if it refers to a known contact, regardless of whether that contact can be found in the "*rcs*", "*rcs_basic_spi_only*", "*rcs_poll*" or "*rcs_poll_basic_spi_only*" lists of the Shared XDMS's "*resource-lists*" document (see section 3.7.4.5.2).

3.7.4.5 Subscriptions and Authorization

3.7.4.5.1 Overview

Presence invitations are subject to reactive authorization to guarantee user privacy. This will allow the invited user (presentity) to accept, block or ignore an invitation to establish a presence relationship.

The presence authorization for basic social presence information shall be symmetric. This means the inviting user automatically authorizes the invited user to see their basic social presence information. The invited user by accepting the presence invitation request both authorizes the inviting user to see their basic social presence information and subscribes to the inviting users presence information.

The RCS presentity shall be able to configure the presence authorization rules, which require the support in the RCS client and in the RCS Presence Server of [PresenceXDM]. The RCS client shall store a presence authorization document that follows [PresenceXDM] and the template rules described in section 5.8 in [PRESENCEIG].

In order for a presentity to be able to authorize the subscription of a watcher, the presentity needs to know which watcher(s) are trying to subscribe to the presence of the presentity.

The RCS client and the Presence Server shall thus support section 5.3.1 and 5.4.4 of [Presence].

When the subscription is authorized successfully, the Presence Server sends the presentity's presence document to the watcher by using the NOTIFY method as defined in [Presence]. The format of the presence notification follows the Presence Data Model as describe above and it contains the information the watcher is allowed to see according to the configured presence rules.

The contacts with whom the RCS user share presence information can be defined as either VIP contacts or non-VIP contacts (see section 3.7.1.4.9). For VIP contacts, presence information changes are received in real time, using a subscription to the corresponding "VIP contacts" buddy list in RLS. For non-VIP contacts the client will poll the corresponding "non-VIP contacts" list in RLS to retrieve presence information changes.

Contrary to the general concept for basic social presence information sharing the authorization for location information is not necessarily mutual: User A can get the location information from User B without having to provide his location information. Furthermore, the user can control whether the information that he/she is capable of sharing social presence information is public or not.

3.7.4.5.2 XML Document Structure

The Presence XDMS shall contain the following authorization rules following, where possible, the recommendations in [PRESENCEIG]:

- "*allow own*" rule – allows subscriptions to own presence data
- "*confirm unlisted*" rule – allows reactive authorization for contacts not yet allowed or blocked
- "*blocked contacts*" – contains those contacts that the user has blocked (points to "*blocked contacts*" list in Shared XDMS)
- "*granted contacts*" rule – will be used as the rule to provide all social presence information (that is, the Basic Social Presence Information and geolocation information)
- "*basic_spi_only_granted_contacts*" rule – will be used by the contacts with whom no location information is being shared.

The RLS XDMS shall for an RCS user contain two entries; one referencing the "*oma_buddylist*" list and one referencing the "*rcs_poll_buddylist*" list, both in Shared XDMS for which the template is described in section 2.14.1. The service URI referencing the "*oma_buddylist*" allows subscribing with one RLS subscription to the presence information of both the VIP contacts with whom only social presence information is shared and those VIP contacts that are also allowed to see the location information. The RCS client will at start-up subscribe to changes to this list by issuing a SUBSCRIBE request to the RLS targeting this list with an expire value >0 (pre-configured in client).

In addition to information on the VIP contacts, the service URI referencing the "*rcs_poll_buddylist*" (see section 2.14.1 for the template) allows the RCS client with one subscription request to retrieve presence information also from the non-VIP contacts with whom only social presence information is shared and those non-VIP Contacts that are also allowed to see the location information. The RCS client will, only on user request or also on regular basis issue a "poll" SUBSCRIBE (that is with expires=0) to this list to obtain the presence information for the contacts in this list.

The maximum amount of poll operations on the non-VIP Contacts buddy list during a certain time period can in the client be configured subject to Service Provider policies (see Annex A).

The Shared XDMS (see section 2.14.1 for the template) shall contain the following lists that are used for presence and are provided and managed by the RCS client:

- "*rcs*" list: This list includes all VIP contacts with which basic Social Presence and location information is shared. Commonly referred in RCS from both the "*oma_buddylist*" and "*oma_grantedcontacts*" lists as the contacts that are allowed to see your presence are also your buddies (symmetric).
  To provide the behaviour described in section 3.7.4.3.3, the "*rcs*" list will contain the own identity of the user. The client shall not allow the user to remove that entry.
- "*rcs_basic_spi_only*" list: This list includes all VIP contacts with which only basic Social Presence information is shared. Commonly referred in RCS from both the "*oma_buddylist*" and "*rcs_basic_spi_only_granted_contacts*" lists as the contacts that are allowed to see your presence are also your buddies (symmetric).
- "*rcs_poll*" list: This list includes all non-VIP contacts with which basic Social Presence and location information is shared. Commonly referred in RCS from both the "*rcs_poll_buddylist*" and "*oma_grantedcontacts*" lists as the contacts that are allowed to see your presence are also your buddies (symmetric). As a difference with the "*rcs*" list, the "*rcs_poll*" list will not contain the own identity of the user.
- "*rcs_poll_basic_spi_only*" list: This list includes all non-VIP contacts with which only basic Social Presence information is shared. Commonly referred in RCS from both the "*rcs_poll_buddylist*" and "*rcs_basic_spi_only_granted_contacts*" lists as the contacts that are allowed to see your presence are also your buddies (symmetric).
- "*oma_buddylist*" list: Contains a reference to the "*rcs*" and the "*rcs_basic_spi_only*" lists where the actual VIP Contacts (or buddies) are stored. The "*oma_buddylist*" is explicitly used from the RLS document.
- "*rcs_poll_buddylist*" list: Contains a reference to the "*rcs_poll*" and the "*rcs_poll_basic_spi_only*" lists where the actual non-VIP Contacts are stored. The "*rcs_poll_buddylist*" is explicitly used from the RLS document.
- "*oma_grantedcontacts*" list: This list includes all contacts you have authorized to see your basic social presence and location information. Contains a reference to the "*rcs*" and "*rcs_poll*" lists.
- "*rcs_basic_spi_only_grantedcontacts*" list: This list includes all contacts you have authorized to see only your basic social presence information. Contains a reference to the "*rcs_basic_spi_only*" and the "*rcs_poll_basic_spi_only*" lists
- "*oma_blockedcontacts*" list: Contains a reference to the "*rcs_blockedcontacts*" list where the actual permanently blocked contacts are stored and to the "*rcs_revokedcontacts*" list with the revoked users that are temporarily being blocked.
- "*rcs_blockedcontacts*" list: Contains all permanently blocked contacts
- "*rcs_revokedcontacts*" list: Contains all revoked contacts that are currently being blocked.

  NOTE1: The "*rcs_revokedcontacts*" list is not intended to be shown to the end user. It is managed automatically.

  NOTE2: A contact should only be in one of the lists used for presence. To ensure this, the RCS client shall check the other lists for an occurrence of the contact when adding it to a list. If the contact occurs somewhere else, the client will remove that entry. A contact will always be added to the new list before being removed from the old one. This applies both when removing a presence relation (see section 3.7.4.5.4) and when changing a contact from

being a VIP Contact to a being a non-VIP Contact or vice versa (see section 3.7.4.5.6).

For RCS, the template definitions below will be used for the different XDM documents related to presence subscriptions and authorizations.

Presence XDMS:
AUID: org.openmobilealliance.pres-rules
Document name: pres-rules
Template

```
<?xml version="1.0" encoding="UTF-8"?>
<cr:ruleset
    xmlns:ocp="urn:oma:xml:xdm:common-policy"
    xmlns:op="urn:oma:xml:prs:pres-rules"
    xmlns:pr="urn:ietf:params:xml:ns:pres-rules"
    xmlns:cr="urn:ietf:params:xml:ns:common-policy">
    <cr:rule id="wp_prs_allow_own">
        <cr:conditions>
          <cr:identity>
              <cr:one id="tel:+1234578901"/>
          </cr:identity>
        </cr:conditions>
        <cr:actions>
          <pr:sub-handling>allow</pr:sub-handling>
        </cr:actions>
        <cr:transformations>
          <pr:provide-services>
              <pr:all-services/>
          </pr:provide-services>
          <pr:provide-persons>
              <pr:all-persons/>
          </pr:provide-persons>
          <pr:provide-devices>
              <pr:all-devices/>
          </pr:provide-devices>
          <pr:provide-all-attributes/>
        </cr:transformations>
    </cr:rule>

    <cr:rule id="wp_prs_unlisted">
        <cr:conditions>
          <ocp:other-identity/>
        </cr:conditions>
        <cr:actions>
          <pr:sub-handling>confirm</pr:sub-handling>
        </cr:actions>
    </cr:rule>
    <cr:rule id="wp_prs_grantedcontacts">
        <cr:conditions>
          <ocp:external-list>
              <ocp:entry anc="http://xcap.gsma.org/resource-
              lists/users/sip:1234578901@gsma.org/index/~~/resource-
              lists/list%5B@name=%22oma_grantedcontacts%22%5D"/>
          </ocp:external-list>
        </cr:conditions>
        <cr:actions>
          <pr:sub-handling>allow</pr:sub-handling>
```

```
        </cr:actions>
        <cr:transformations>
          <pr:provide-services>
              <pr:all-services/>
          </pr:provide-services>
          <pr:provide-persons>
              <pr:all-persons/>
          </pr:provide-persons>
          <pr:provide-devices>
              <pr:all-devices/>
          </pr:provide-devices>
          <pr:provide-all-attributes/>
        </cr:transformations>
    </cr:rule>

    <cr:rule id="rcs_basic_spi_only_grantedcontacts">
        <cr:conditions>
          <ocp:external-list>
              <ocp:entry anc="http://xcap.gsma.org/resource-
              lists/users/sip:1234578901@gsma.org/index/~~/resource-
              lists/list%5B@name=%22rcs_basic_spi_only_grantedcontacts%22%5D"/>
          </ocp:external-list>
        </cr:conditions>
        <cr:actions>
          <pr:sub-handling>allow</pr:sub-handling>
        </cr:actions>
        <cr:transformations>
          <pr:provide-services>
              <pr:all-services/>
          </pr:provide-services>
          <pr:provide-persons>
              <pr:all-persons/>
          </pr:provide-persons>
          <pr:provide-devices>
              <pr:all-devices/>
          </pr:provide-devices>
          <pr:provide-note>true</pr:provide-note>
          <pr:provide-status-icon>true</pr:provide-status-icon>
          <pr:provide-unknown-attribute
              ns="urn:oma:xml:pde:pidf:ext:1.1"
              name="link">
              true
          </pr:provide-unknown-attribute>
          <op:provide-willingness>true</op:provide-willingness>
          <pr:provide-unknown-attribute
              ns="urn:oma:xml:prs:pidf:oma-pres"
              name="service-description">
              true
          </pr:provide-unknown-attribute>
        </cr:transformations>
    </cr:rule>

    <cr:rule id="wp_prs_blockedcontacts">
        <cr:conditions>
          <ocp:external-list>
              <ocp:entry anc="http://xcap.gsma.org/resource-
              lists/users/sip:1234578901@gsma.org/index/~~/resource-
```

```
            lists/list%5B@name=%22oma_blockedcontacts%22%5D"/>
          </ocp:external-list>
        </cr:conditions>
        <cr:actions>
          <pr:sub-handling>block</pr:sub-handling>
        </cr:actions>
    </cr:rule>
</cr:ruleset>
```

Table 72: Presence Rules Template

NOTE: If the client is configured to use a presence based capability discovery (as described in section 2.6.1.2, the *rcs_allow_services_anonymous* rule described in Table 31 should be included in this template.

RLS XDMS:
AUID: rls-services
Document name: index
Template:

```
<?xml version="1.0" encoding="UTF-8"?>
<rls-services xmlns="urn:ietf:params:xml:ns:rls-services">
    <service uri="sip:1234578901@gsma.org;pres-list=rcs">
        <resource-list>http://xcap.gsma.com/services/resource-
        lists/users/sip:1234578901@gsma.org/index/~~/resource-
        lists/list%5B@name=%22oma_buddylist%22%5D</resource-list>
        <packages>
          <package>presence</package>
        </packages>
    </service>
    <service uri="sip:1234578901@gsma.org;pres-list=rcs_poll">
        <resource-list>http://xcap.gsma.com/services/resource-
        lists/users/sip:1234578901@gsma.org/index/~~/resource-
        lists/list%5B@name=%22rcs_poll_buddylist%22%5D</resource-list>
        <packages>
          <package>presence</package>
        </packages>
    </service>
</rls-services>
```

Table 73: Presence RLS Template

NOTE: the lists in the Shared XDMS and the general procedures on the handling of XDM requests and the creation of the documents are described in section 2.14.

3.7.4.5.3 Client Procedures, Initiation of Presence Sharing

When initiating a presence sharing request, the inviting user's RCS client adds the invited user's URI to the "*rcs*" list in Shared XDMS according to the procedures in [Shared-XDM].

When the invited user receives a notification to establish a presence relation, the user can either:

1. Accept the invitation, whereas the RCS client of the invited user adds the inviting User's URI to the "*rcs*" list in Shared XDMS (see section 2.14.1) according to the procedures in [SHARED-XDM].
2. Block the invitation, whereas the RCS client of the invited user adds the inviting User's URI to the "*rcs_blockedcontacts*" list in Shared XDMS (see section 2.14.1) according to the procedures in [SHARED-XDM].

3. Ignore the invitation, whereas the RCS client of the invited user removes the presence sharing invitation.
4. Not answer the invitation. The presence sharing invitation is pending in the client until either “accepted” (case 1), “blocked” (case 2) or “ignored” (case 3). In the signalling, there is no difference from the “ignore” case.

3.7.4.5.4 Client Procedures, Removal of Presence Sharing

When the user decides to end the presence relationship with one of their contacts, they have to use the revoke option on their device. This triggers a notification to the user as defined in section 3.7.1.4.6 asking for confirmation. When this is confirmed, the client will put the user on the “*rcs_revokedcontacts*” list, subsequently remove the user from the “*rcs*” or “*rcs_basic_spi_only*”, “*rcs_poll*” or “*rcs_poll_basic_spi_only*” list and remove the contact’s presence information from the cache as defined in section 3.7.4.4.1. When putting an entry for the contact in the “*rcs_revokedcontacts*” list the client includes a last modified attribute that indicates the current time in UTC.

When a client notices it has been blocked by a contact with whom Social Presence was shared (that is the RLS notify indicates the subscription is in state “*terminated*” and the reason indicates “*rejected*”), it will remove the contact from the “*rcs*” or “*rcs_basic_spi_only*”, “*rcs_poll*” or “*rcs_poll_basic_spi_only*” list and remove the contact’s cached presence information. Note that for a non-VIP contact (in the “*rcs_poll*” or “*rcs_poll_basic_spi_only*” list) there could be a delay in the detection of this change.

All clients will process the “*rcs_revokedcontacts*” list periodically and remove those contacts that have been included in the list for a sufficiently long period already (for example several days). For that they will compare the last-modified attribute of the entries to the current time. Both the interval at which the list is checked (REVOKE TIMER) and the period that a contact should remain in this list (REVOKE TIMER) is a Service Provider configurable client parameter defined in Annex A.

With regards to the communication capabilities both clients should fall back to the procedures as defined in section 2.6 for sharing of capabilities between contacts not sharing social presence information.

3.7.4.5.5 Conditional Event Notification

The support of conditional event notification is strongly recommended for the clients (i.e. Watcher and Watcher Information Subscriber) and for the servers (i.e. Presence Server and RLS) to optimize presence traffic at UNI and NNI.

An RCS client should support subscription with conditional event notification, as defined in section 5.2.6 and section 5.3.2 of [Presence2.0_TS].

An RCS RLS should support subscription with conditional event notification, as defined in section 5.2 of [Presence2.0_RLS_TS].

An RCS Presence Server should support notification with conditional event notification, as defined in section 5.5.3.8, 5.5.3.9 and 5.5.4.2 of [Presence2.0_TS].

An RCS RLS should support notification with conditional event notification, as defined in section 5.4 of [Presence2.0_RLS_TS].

3.7.4.5.6 Client Procedures, managing of VIP and non-VIP Contacts

When the user decides to change a user from being a VIP Contact to being a non-VIP Contact (or vice versa) the client will first add the user’s URI to the target list and after this, remove the user’s URI from the list where it was previously stored. That is, when changing a user from being a VIP Contact to a non-VIP Contact, the client will first add the user’s URI to the “*rcs_poll*” (if previously in the “*rcs*” list) or “*rcs_poll_basic_spi_only*” list (if previously in the “*rcs_basic_spi_only*” list) and then remove the URI from the “*rcs*” or

"*rcs_basic_spi_only*" list respectively. When changing a user from being a non-VIP Contact to a VIP Contact, the client will first add the user's URI to the "*rcs*" list (if previously in the "*rcs_poll*" list) or "*rcs_basic_spi_only*" list (if previously in the "*rcs_poll_basic_spi_only*" list) and then remove the URI from the "*rcs_poll*" or "*rcs_poll_basic_spi_only*" list respectively.

3.7.4.5.7 Multidevice Handling

Any negative effects of XDM document changes in a multidevice context are countered through the XDM document handling as it is described in section 2.14.2.

Several situations should be dealt with:

- The user owning multiple clients is invited by a contact to share social presence information.
  All the user's active clients will receive watcher information notifications both when the contact subscribes for the user's social presence information (subscription entering the "*pending*" state) and when the user accepts or blocks the "invitation" on one of their clients (subscription going out of the "*pending*" state). When the user accepts the invitation on one of their clients, the other clients will also start receiving the social presence information of the contact.
- The user owning multiple clients invites a contact to share social presence information from one of their clients.
  In this case their other clients will receive presence notifications indicating that a subscription to the contact entered the pending state and notifications including the other user's social presence information when the contact accepted the "invitation". If the contact blocks the "invitation", there will be presence notifications to all the user's clients indicating that the subscription was terminated. The clients shall use these unexpected notifications as triggers to update the locally stored copy of the Shared XDMS's "*resource-lists*" document if they cache that kind of information locally.
- The user revokes the presence sharing with a contact from one of their clients.
  Again his other clients that are online will receive unexpected presence notifications indicating that the subscription to the contact's social presence information was terminated. If they cache the information in the Shared XDMS's "*resource-lists*" document locally, they shall use this notification as a trigger to verify that the information is still up-to-date.
  Changes are done while the client was offline. A client that caches the information in the Shared XDMS's "*resource-lists*" document locally should check whether that document has changed when it comes online. Therefore, this will not cause any issues.
- The user owning multiple clients changes a contact from being a VIP contact to being non-VIP contact from one of their clients. His other clients that are online will receive unexpected presence notifications indicating that the subscription to the contact's social presence information was terminated. If they cache the information in the Shared XDMS's "*resource-lists*" document locally, they shall use this notification as a trigger to verify that the information is still up-to-date.
- The user owning multiple clients changes a contact from being a non-VIP contact to being a VIP contact from one of their clients. In this case, their other clients will receive presence notifications indicating that a subscription to the contact has been created and notifications including the other user's social presence information. Again, the clients shall use these unexpected notifications as triggers to update the locally stored copy of the Shared XDMS's "*resource-lists*" document if they cache that kind of information locally.

3.7.4.6 RLS Server Handling

3.7.4.6.1 Nickname Handling

A RLS server supporting RCS shall include any display name it received in the *P-Asserted-Identity* and *From* headers of the RLS subscription in the corresponding header of the related backend subscriptions that it sends to the Presence Server.

3.7.4.7 Presence Server Handling

3.7.4.7.1 Nickname Handling

A Presence Server supporting RCS shall include any display name it received in the *P-Asserted-Identity* header field of a presence subscription in the display-name attribute of any entry related to that subscription in the *application/watcherinfo+xml* body that is sent to the clients of the served RCS presentity that was the target of the subscription. If the *P-Asserted-Identity* header field does not contain any display name, the display name provided in the *From* header field of the subscription will be used, if any.

3.7.4.8 XDM Server Handling

3.7.4.8.1 Status Icon

In the network the retrieval of the information referred to by the link to the status icon will be realized in an architecture as described in [XDM1.1_AD] with the addition of the Cross-Network Proxies and XDM-8 and NNI-1 interfaces defined in [XDM2.0_AD]. The required functionality of the Cross-Network Proxy is limited to the authorization, data transfer and routing of XCAP functionalities. The routing of search requests is not applicable to RCS. For RCS the supported protocols on the NNI-1 interface are limited to XCAP, "*limited XQuery over HTTP*" is not supported.

At the functionality level, this means that the identity provided by the Aggregation Proxy is not only shared on the XDM-4 and enabler specific reference points between the Aggregation Proxy and the Enabler specific XDMS as it is described in [XDM1.1_Core] section 6.4.1, but also on the XDM-8 and NNI-1 interfaces as it is described in [XDM2.0_Core] section 5.1.3. The Integrity and Confidentiality protection of [XDM1.1_Core] section 6.4.2 is extended to the NNI-1 interface as it is described in [XDM2.0_Core] section 5.1.4. Furthermore in addition to the functionality described in [XDM1.1_Core], the Aggregation Proxy shall route requests to the Cross-Network proxy as it is described in [XDM2.0_Core] section 6.3.1.1 and route the Cross-Network Proxy's responses back to the XDM client. The procedures for routing requests to the search proxy that are described in [XDM2.0_Core] section 6.3.1.1 are not applicable for RCS. Finally the functionality of the Cross-Network Proxy as it is described in [XDM2.0_Core] section 6.5 and subsections shall be supported with the exception of all functionality related to the routing of Search Requests and Search Responses.

3.7.5 NNI and IOT considerations

The NNI interfaces for SPI sharing shall behave according to the procedures described in section 2.12 and the documents it refers to.

3.7.6 Implementation guidelines and examples

3.7.6.1 SPI transaction handling

Initiator side

1. An RCS user that wants to Share SPI with a contact selects the contact entry in their local enriched address book.
2. They select in the menu "share" (if available, that is the contact has the SPI service capability) the function "Share Social Presence" and can see by using the SPI general

menu the SPI status associated with the contact (“idle”, “pending” “activated”, “terminated”)

3. This SPI general menu, depending on the SPI status, enables them to invite the contact to share SPI with following options
   - “VIP contact”: YES / NO (default NO)
   - “Authorize Location Sharing”: YES / NO (default NO)

   NOTE: at any moment, for these 2 options, when the SPI status becomes “active”, the general Share SPI menu offers the user the possibility to change their choices

   - “Nickname” text field: free user text
4. Then the user can follow the SPI status evolution the SPI status by selecting the contact and activating the SPI general menu
   - “pending”: the contact has not yet accepted to share SPI with them
   - “active”: the contact has accepted to share SPI with them
   - “terminated”: The contact, after acceptation, has decided to revoke sharing Presence Information

Callee side

1. The RCS user is triggered by a pop up SPI menu that a distant user has invited them to share their Social Presence Information
   - If the user already has a contact entry for the inviting user in the local address book, then the name assigned to the contact entry in the local address book of the user appears in this SPI menu
   - If the user is not present in the local address book, then the “nickname” of the inviting user (if any provided) and their E.164 address appear in the menu instead
2. The SPI pop up menu proposes allows actions through buttons and fields to be filled
   - “Accept”: YES/NO
   - “VIP contact”: YES / NO (default NO)
   - “Authorize Location Sharing”: YES / NO (default NO)

   NOTE: at any moment, for the latter 2 options, when the SPI status becomes “active” the general Share SPI menu offers the user the possibility to change their choice

3. Then the user can follow the SPI status evolution by selecting the contact and activating the SPI general menu
   - “active”
   - “terminated”: The contact, after acceptation, has decided to revoke sharing Presence Information

SPI status “active”

At any moment, in the “active state” the user can choose for a contact selected in the address book:

- To modify SPI sharing parameters: VIP contact, Geolocation Sharing authorisation
- To revoke SPI sharing

3.7.6.2 Availability handling

The user can choose how they appear to their contacts: “Available” or “Not Available”.

3.7.6.3 Free Text handling

The user enters some free text possibly including emoticons. They are blocked when the length of the text reaches the limit fixed by the Service Provider.

3.7.6.4 Icon handling

The user is asked to choose an image in the local file system of the device from a sub set of the images that are candidate to be part of the user SPI (filter based on file size: the size of the icon must not exceed what is authorized by the Service Provider).

3.7.6.5 URL label

The user is asked to enter a URL and an associated free text. The user may be assisted by the application to enter the information depending on the Service Provider settings.

3.7.6.6 Geolocation handling

In a manual mode, user manually picks a position (x, y) on a map or user requests for an update of their position (x, y) information. Then, geolocation information is given by RCS client towards authorized enriched contacts as soon as it has been made available on the RCS client by the user.

In automatic mode, update of location coordinate information (x, y) is automatically made and given to the authorized enriched contacts on a regular basis.

Manual mode and automatic mode are further detailed below.

3.7.6.6.1 Display Modes

Three displays modes are possible:

1. Text: a user is located and the result is given to their authorized enriched contacts under a declarative text format (Paris, La Défense). The declarative text is always manually edited by the user.
2. Map: a user is located and the result is given as coordinate information (x, y) to his authorized enriched contacts and displayed under a map format. When the user is displayed as a dot on a map, their location information can also be displayed as text in other screens. For example, if a user has updated his location to a position in the centre of London on a map, some screens without a map may display his location using the declarative text edited by the user (for example, “London, UK”).
3. A combined display of text and a map

3.7.6.6.2 Update Information

Declarative location text information is always manually edited/updated by the user.

The Geolocation information update regarding coordinate information (x, y) can be either:

- Manual
    - The user can select their location manually on a map, by either entering text that is then processed to provide location (as coordinate information [x,y]) on a map (for example Google Maps) or, for example, by dragging and dropping a “pin” on a map to the desired location. This user-chosen location can be different from the user’s actual location.
    - Triggering their actual current location (based, for example, on a GPS signal from the device or a mobile network-based location). For example, they click on the location update button, and coordinate information (x,y) is automatically filled)

- Automatic
  - (User A decides that they want their authorized contacts to be informed regarding their coordinate position (x,y) on a regular basis). Location coordinate information (x, y), and any update is automatically made and given to authorized enriched contacts on a regular basis.

Other recommendations for implementation from the end user's perspective (these are only meant as examples and not actual specifications):

- For Fully Automatic update, the user shall be able to choose the level of accuracy for their location
  - Country
  - City
  - Street (most accurate location)
- In addition to having a map displayed per contact inside the address book (at -1 or -2 navigation levels), there might be the possibility to have a consolidated map with all contact location information (within the scope defined : country, city or street). The starting position of the map is the user's current position, if available. See also section 3.10.

3.8 IP Voice Call

3.8.1 Feature description

This feature provides an IP Voice Call service on an RCS device. An IP Voice Call interoperates with other RCS devices including VoLTE/VoHSPA as defined in [PRD-IR.92] and [PRD-IR.58] and with CS/PSTN (Public Switched Telephone Network) voice calls. The voice call is provided via IP Voice when the access network allows it, and may be provided via CS voice when IP Voice is not available, depending on the device and network capabilities.

The minimum set of supplementary services provided is described in [PRD-IR.92].

At any time, either user can terminate the IP Voice Call.

3.8.2 Interaction with other RCS features

3.8.2.1 Interaction with RCS Messaging, File Transfer, Content Sharing, Geolocation PUSH

IP Voice Call must use a separate SIP session which is not shared with Standalone messaging (section 3.2), Chat (section 3.3), Group Chat (section 3.4), File Transfer (section 3.5), Content Sharing (section 3.6) or Geolocation PUSH (section 3.10). Interaction with Content Sharing is covered in section 3.6.2. Interaction with Video Call is covered in section 3.9.2.

3.8.2.2 Interaction with CS Voice (Telephony)

In RCS, voice services are supported both on access networks natively offering CS Voice telephony (thus without the need for CS/PS voice conversion by Media Gateway components in the core network) as well as on IP based access networks using PS voice.

Within the RCS multi-device context, any of the user's clients could be receiving the final leg of the voice service in any of these ways:

- through the CS network (i.e. for a device in RCS-CS mode) as CS Voice telephony,
- through the IMS network using PS voice (i.e. for devices in RCS-LTE, RCS-HSPA or RCS-AA mode and even for devices in RCS-CS mode if allowed by the service provider).

For NNI, the interconnection between networks could use CS voice service NNI or could use PS voice. For both the UNI and NNI interfaces, several options are possible to achieve the required behaviour, and neither choice is optimal in all circumstances. Therefore, this choice is considered out of scope for RCS.

3.8.3 High Level Requirements

3-8-1 The scope of the requirements for IP Voice Call are those found in [PRD-IR.92] and [PRD-IR.58].

3.8.4 Technical Realization

At a technical level the voice call service shall be based on [PRD-IR.92] and [PRD-IR.58].

Since in RCS a user may register a primary and one or more secondary devices in IMS, incoming SIP requests are forked. This principle also applies to the case where the user has several SIMs assigned to the same phone number (i.e. the same IMS subscription), and consequently, incoming SIP requests are forked.

This also applies to incoming SIP requests for IP Voice Calls, so it is expected that they be forked in the same way as other RCS related SIP requests are forked, i.e. in parallel. For voice sessions set up according to [PRD-IR.92] and [PRD-IR.58], the support for early media as described in [PRD-IR.92] and [PRD-IR.58] is required.

Broadband Access clients which support and are configured for RCS IP Voice Call but are not enabled for VoLTE/VoHSPA (and therefore do not make use of the IMS APN as specified in section 2.9.1.4) shall behave as a device in RCS-AA mode as defined in section 2.2.1. When using this service, the device in RCS-AA mode shall clearly indicate to the user that this is not a telephony replacement service.

If the service is enabled by Service Provider policy settings (*PROVIDE RCS IP VOICE CALL* as defined in section A.1.14), the same applies for devices in RCS-CS mode supporting the RCS IP Voice Call as defined in section 2.2.1, and again using this service, the device in RCS-CS mode shall clearly indicate to the user that this is not a telephony replacement service. Furthermore the continuity of such an RCS IP Voice Call relies on the continuity of the IP connectivity (i.e. SR-VCC does not apply).

There is no need to re-register when entering/leaving LTE/HSPA coverage.

3.8.4.1 Devices enabled for VoLTE/VoHSPA

If the domain selection has selected 3GPP PS access for voice (VoLTE/VoHSPA) this access is used for RCS features as well. If either VoLTE and/or VoHSPA is supported any of these is assumed to be natively implemented and integrated within the device. The IMS registration shall be shared between VoLTE/VoHSPA and RCS.

3.8.4.2 Devices using CS domain for voice calls

A device may use the CS domain possibly via Circuit Switched Fallback (CSFB) for voice calls when it is not enabled for VoLTE/VoHSPA, or it is enabled for VoLTE/VoHSPA but the current network does not support VoLTE/VoHSPA (e.g. the serving network does not support VoLTE or an IMS roaming agreement is not in place).

LTE access can be used for RCS features providing there is no ongoing CS call.

LTE devices not enabled for VoLTE will fall back to CS for voice calls. Once CS fallback occurs, LTE access is dropped, and RCS functionality is provided via 3G/2G access.

3.8.4.3 Devices using RCS IP voice calls

A device/client supporting and configured to use RCS IP Voice Calls shall indicate this in SIP INVITE requests and responses according to Table 3. The device/client shall also

include a *P-Preferred-Service* header field with the MMTEL ICSI as per [PRD-IR.92] and include the relevant subclass, i.e.,

```
P-Preferred-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall
```

The use of the media feature tag *+g.gsma.rcs.ipcall* allows the Service Provider to differentiate between regular MMTEL (VoLTE/VoHSPA) service (without this additional media feature tag) and RCS IP voice service (with this additional media feature tag) which indicates a strong preference for no breakout (IP end to end).

3.8.4.4 Flows

Since the voice call UX is well-known, it is not necessary to provide basic message flows and a reference UX. A flow is provided for handling of an incoming CS call when there is already an IP Voice Call.

3.8.4.4.1 Incoming CS Voice call when already in an RCS IP Voice Call

In this scenario an RCS IP Voice Call is ongoing between two users.

User A receives an incoming CS Voice call from User C. User A shall receive an indication of the incoming call. User A shall be able to:

1. Reject the incoming CS Voice call from User C (and thus stay in the same RCS IP Voice Call with User B as long as data connectivity was not lost);
2. Accept the incoming CS Voice call from User C, and consequently tear down the RCS IP Voice Call with User B;
3. Put the RCS IP Voice Call with User B on hold (as long as data connectivity was not lost) and answer the incoming CS Voice call from User C.

   NOTE1: If the device is using LTE and the CS Fallback network is 2G without DTM support, the data connection is suspended and resumed after the CS call, so the RCS IP Voice Call may or may not still be there.

   NOTE2: when a Service Provider's deployment allows directing incoming RCS IP Voice Calls to devices that are in a CS voice call already, equivalent options will be available to the user when an RCS IP Voice Call is received while in a CS call.

3.8.5 NNI and IOT considerations

No specific guidelines apply other than what is already defined in Section 2.12.

3.8.6 Implementation guidelines and examples

From the UX point of view, two possible entry points to the voice service are:

1. Address book/Call-log: A voice call can be initiated with any registered contact – contact oriented initiation.
2. Chat window: From the Chat (one-to-one Chat only) window a voice call can be initiated using the relevant menu item. The experience is identical to the address book/call-log.

Since the voice call UX is well-known, it is not necessary to provide implementation guidelines and examples.

3.9 IP Video Call (IR.94)

3.9.1 Feature description

This feature provides an IP Video Call between two RCS devices with synchronization between the audio and video streams, thus providing lip synchronization. For voice the IP

Voice Call service (as described in section 3.8) is used, along with the clarifications for call establishment described in [PRD-IR.94].

The continuity of an RCS IP Video Call relies on the continuity of the IP connectivity (i.e. SR-VCC does not apply here).

The establishment of the IP Video Call session can be achieved in three possible ways:

1. '**Direct launch**', if no previous voice call was established between the contacts.
2. '**Upgrade to IP Video Call'**, if the users were already engaged with each other in an IP Voice Call communication.
3. '**Replace a CS voice with an RCS IP Video Call**', if the users were already engaged with each other in a CS voice communication. This can only happen on devices that are in RCS-CS mode.
   Any party in the CS call could initiate a capability query message. After completing a successful capability query, the CS voice service may be replaced with the RCS IP Video Call service. However, if the query fails, the operator may choose to allow launching this service anyway.

From the user experience perspective the RCS user can toggle between front camera ("me"), the rear camera ("what I see") and a file (video stream), at any time when using the IP Video Call service.

NOTE: The Video Call service in this context is seen as a superset of Video Share use cases as described in 3.6.1.2 offering lip synch in addition.

In all cases, when invited for a video call an RCS user can either:

- Accept the video call establishing a full duplex video call
- Accept only to receive the inviting user's video content establishing a call where the video part runs in simplex mode alongside a full duplex audio call. In this case the accepting user can at any time decide to move the video part to full duplex as well.
- Accept the call as audio only, i.e. decline the video part of the communication. Voice call is established or continues.
- Decline the video call i.e. no communication is established to any of the receiving user's devices when declining the video call. The call may be redirected to a voice or video messaging system however depending on the policies of the receiving user's network.

When the video stream of the IP Video Call is realised in a full duplex mode, at any time, either user can decide to migrate from a full duplex mode to a simplex mode, i.e. deactivate the sending of their video stream. They can later decide to migrate from a simplex mode to a full duplex mode again.

At any time, either user can terminate the IP Video Call (both audio and video stream or only the video stream).

An RCS device may learn and remember that a contact is IP Video Call capable upon receiving a SIP INVITE request for an IP Voice or IP Video Call. Communication with [PRD-IR.94] compliant devices should not be prevented if RCS procedures for service capability discovery are not supported by those devices.

3.9.1.1 Direct Launch

When both parties support video call at any particular point in time (e.g. by the capability exchange described in section 2.6), either user can initiate the setup of a video call. The receiving user determines whether the call will be initiated in full or simplex mode.

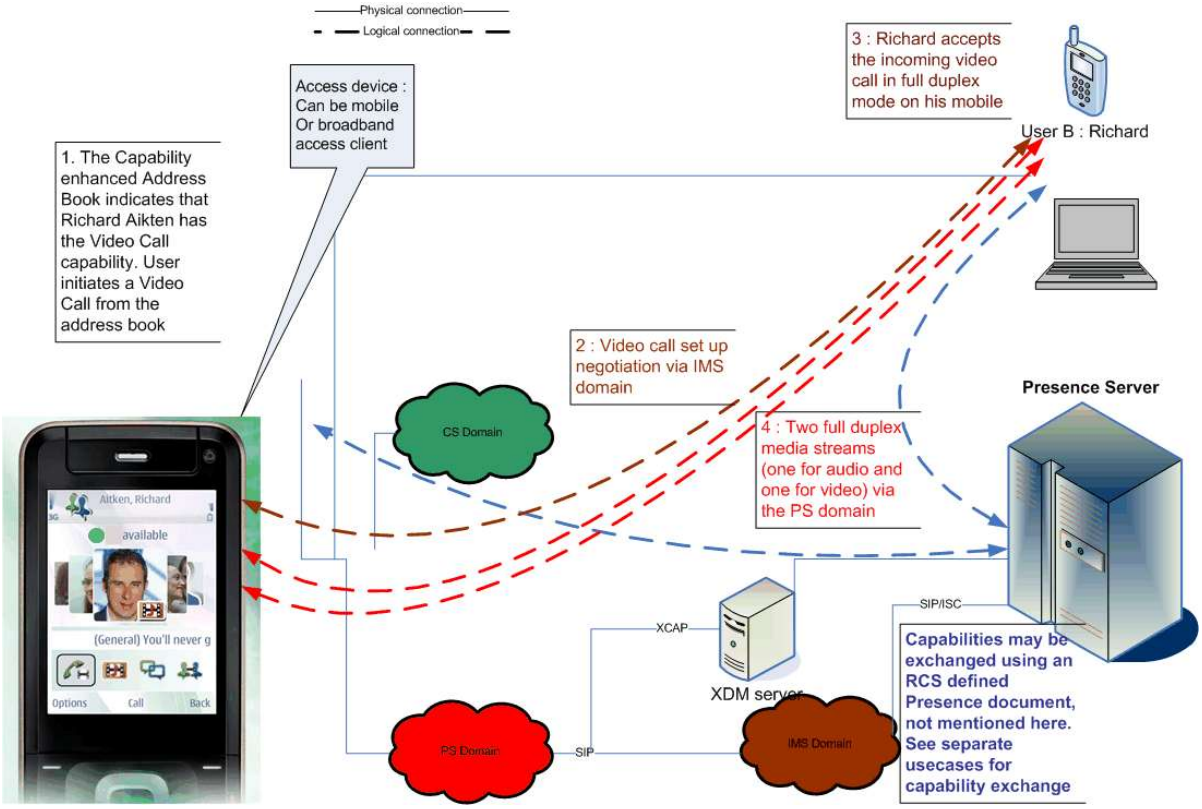


Figure 115: Full duplex video call

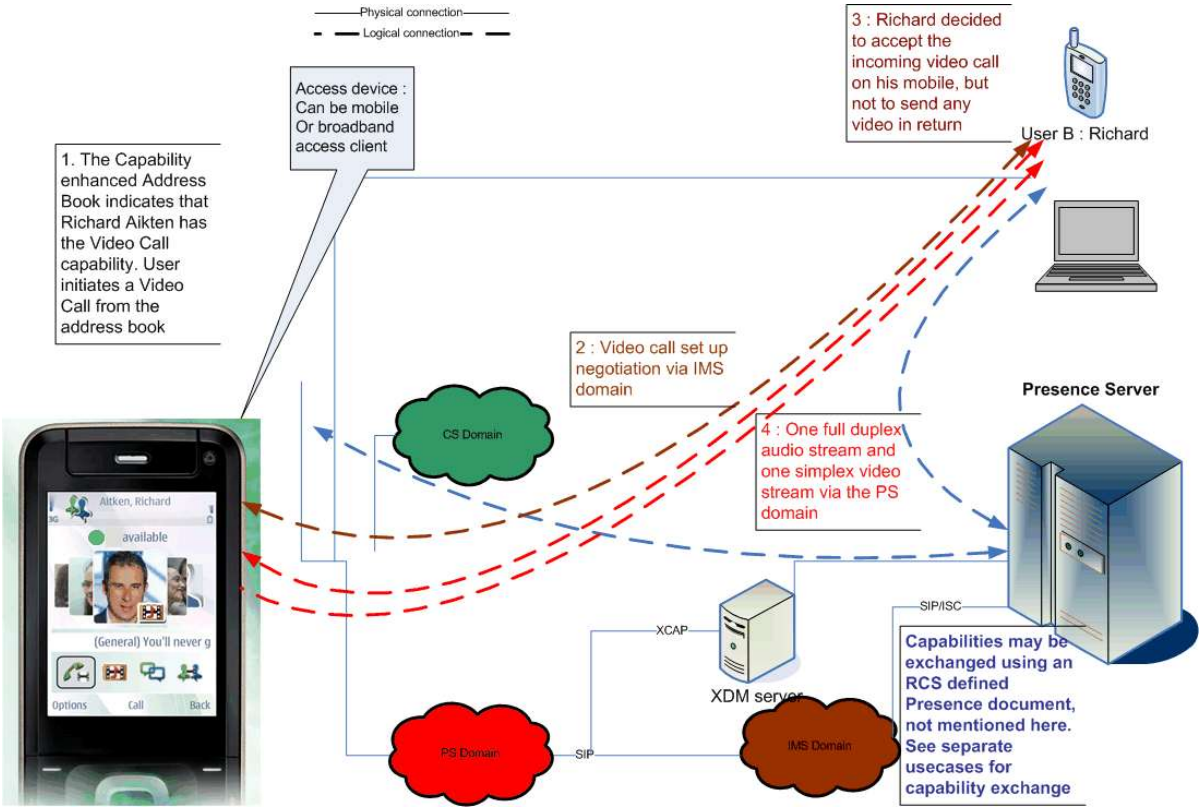


Figure 116: simplex video call

Users could switch between the full duplex and simplex variants of the video during the call. This would result in a new negotiation via the IMS domain for the ongoing call.

NOTE: multiparty calls are also possible.

3.9.1.2 Upgrade to IP Video Call

As stated in section 3.9.1, a user could also start a video call from an existing IP Voice Call (that is the service described in section 3.8).

When the devices on the call all support video call at a particular point in time, either user can initiate the upgrade to a video call by selecting the corresponding option.

If the voice call was entirely (end-to-end) in the PS domain this initiates a negotiation via the IMS domain and if the other user accepts the upgrade a simplex or duplex video stream is added to the ongoing call.

NOTE1: if one end of the call moved to CS, the upgrade may fail, but the voice call would remain in place. If the other party is an RCS user, the party wanting to upgrade may have discovered the fact that video call is no longer available due to the capability exchange described in section 2.6 and should therefore not be offered the possibility to upgrade.

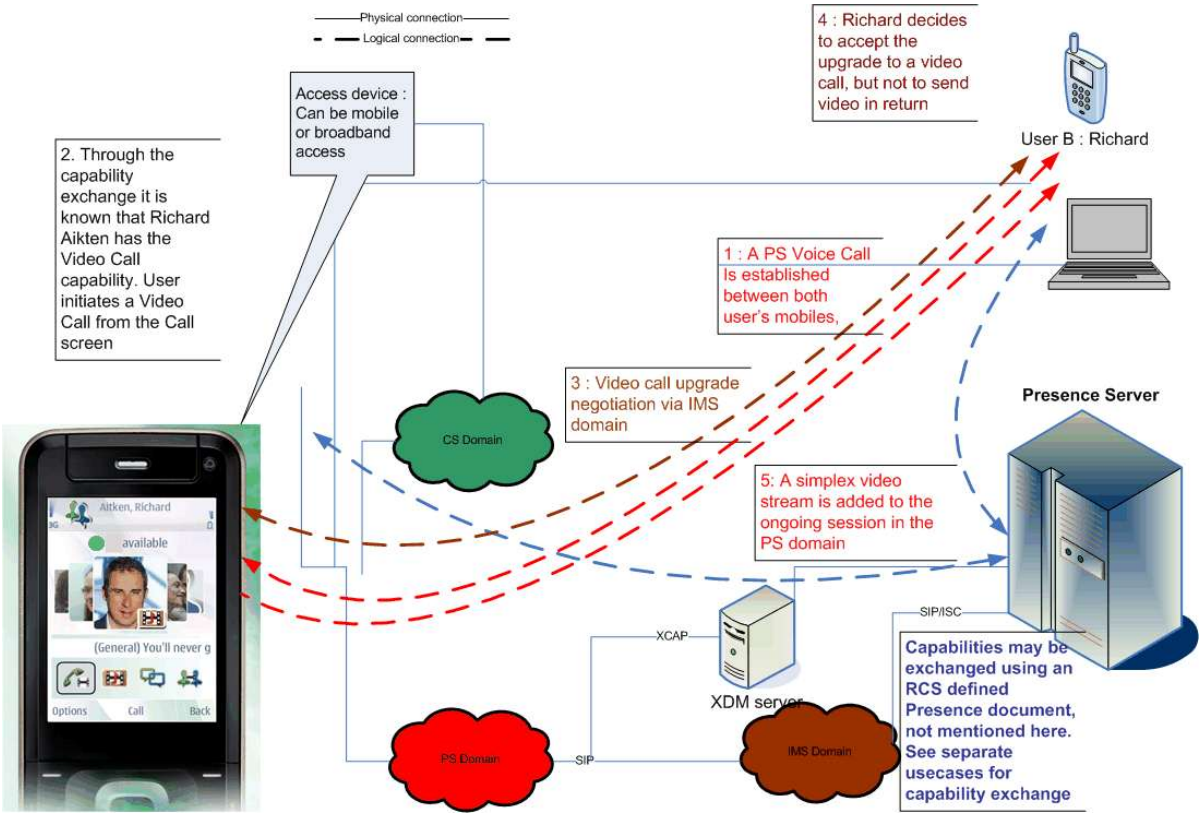


Figure 117: Upgrade PS call to video call

NOTE2: The behaviour is the same for the scenario where the user accepted the video call as a full duplex service.

3.9.1.3 Replace a CS Voice with a RCS IP Video Call

As stated in section 3.9.1, a user can replace a CS Voice call with an RCS IP Video Call. This is accomplished by the client first tearing down the ongoing CS Voice call.

When the devices on the call each support video call at a particular point in time, either user can initiate the upgrade to a video call by selecting the corresponding option, which results in the CS Voice call being terminated and the RCS IP Video Call being initiated.

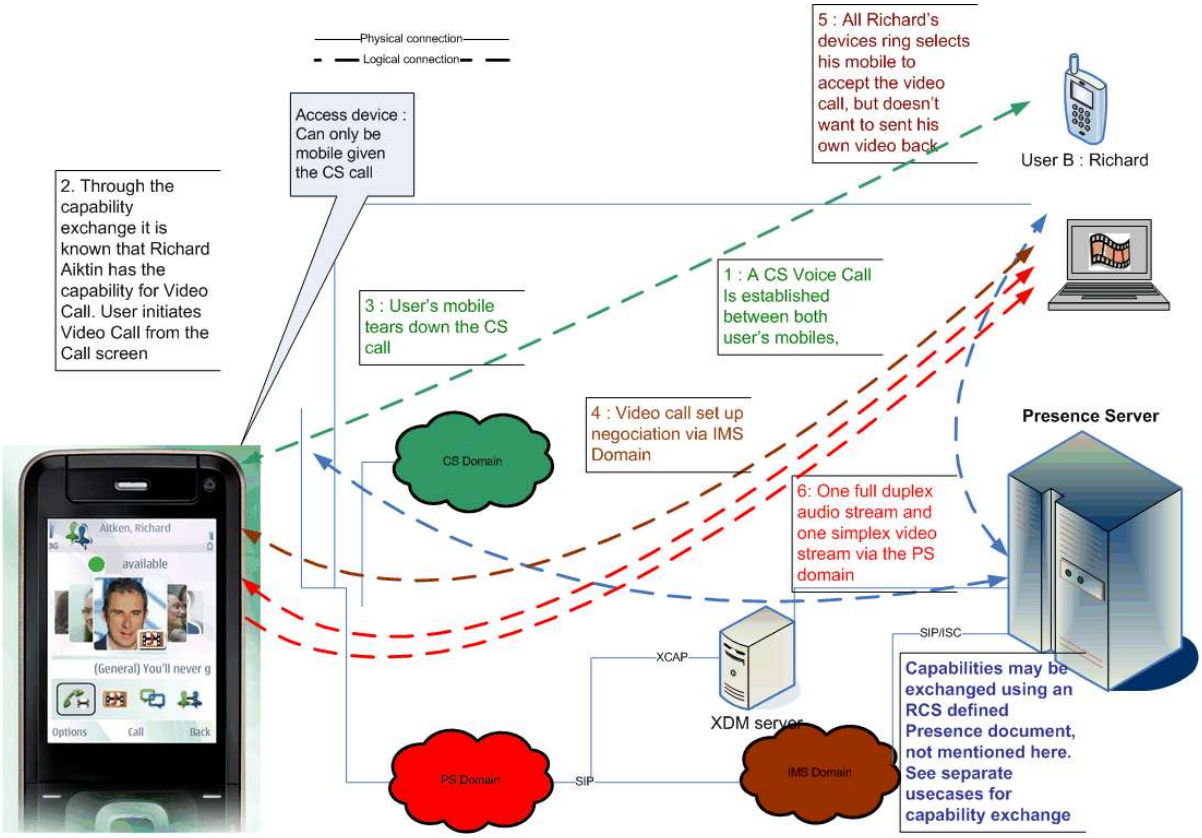


Figure 118: Replace CS call with an RCS IP Video Call

NOTE: The behaviour is the same for the scenario where the user accepted the video call as a full duplex service.

3.9.2 Interaction with other RCS features

3.9.2.1 IP Voice Call

The IP Video call must use the same SIP session as the IP Voice Call (see section 3.8).

The video call service has a strong interaction with the voice call service since both services offer the option for full-duplex real-time communication. That strong relation results in the option to upgrade an existing voice call to a video call as described in section 3.9.1.2. An end-to-end IP Voice Call is upgraded by adding an additional media stream to the ongoing session.

Communication Waiting: when the user is on a voice call and a request for an unrelated video call is received (or vice versa), the device shall handle this video call in the same way as a second voice call coming in. Meaning it will behave differently for the scenario where no call was active and will thus not start ringing loudly and shall use Communication Hold appropriately if the new call is accepted without terminating the ongoing one.

3.9.2.2 Video Share

The IP Video Call and Video Share service capabilities are mutually exclusive: when both ends are capable of using the IP Video Call service (as per [PRD IR.94]), then IP Video Call shall be used as the service to share contents instead of Video Share as described in section 3.6. If one or both ends are not capable of using the IP Video Call service, then Video Share will be used to provide the service. Therefore when performing a capability exchange within a call, if the Video Call capability is set as available, the Video Share capability shall also be made available.

3.9.3 High Level Requirements

3-9-1 In a video call the delay difference between audio and video media shall be unnoticeable (that is lip sync is provided)

3-9-2 The overall delay on both media shall allow for a conversational service

3-9-3 The quality of the video shall be high. At least H.264 level 1.2 shall be supported in suitable circumstances matching the similar requirement in [PRD-IR.94]

3-9-4 It shall be possible to establish a video call without having an active voice call between the parties in the call

3-9-5 It shall be possible to convert an ongoing IP Voice Call (that is as in section 3.8) into an IP Video Call

3-9-6 The receiver shall be able to accept the call in full-duplex mode and in simplex mode in which case no content is sent back to the originating party.

3-9-7 It shall be possible for either party to turn a full duplex video call into a simplex one by terminating the streaming.

3-9-8 If the device has multiple cameras it shall be possible to toggle between them.

3-9-9 The receiver shall be able to reject the video call. This rejection does not affect an ongoing voice call.

3-9-10 Either party shall be able to terminate an active video call

3-9-11 Terminating an active video call shall terminate the communication regardless of whether the call was initiated directly as a video call or initially started as a voice call only.

3-9-12 At least the minimum set of supplementary services defined in [PRD-IR.94] shall be supported

3.9.4 Technical Realization

The IP Video Call service shall be based on [PRD-IR.94]. A device configured for VoLTE/VoHSPA, whether it is in RCS-VoLTE, RCS-VoHSPA or RCS-CS mode as defined in section 2.2.1 shall behave according to the descriptions in [PRD-IR.94]. Broadband access devices shall behave as devices in RCS-AA mode.

When a device in RCS-AA or RCS-CS mode is using this service on a best effort basis, the device shall clearly indicate to the user that they are not using a telephony replacement service. For a device in RCS-CS mode the service will only be available depending on the Service Provider policy settings (*PROVIDE RCS IP VIDEO CALL* as defined in section A.1.14).

Integration of resource management and SIP is done as per [PRD-IR.94] for devices in RCS-VoLTE mode, and as per [PRD-IR.94] and [PRD-IR.58] for devices in RCS-VoHSPA mode. No specific requirements for resource management are required for devices in RCS-AA or RCS-CS mode.

For RTP media and RTCP usage, a device in RCS-AA mode shall follow the requirements for NAT traversal as specified in section 2.8.

3.9.4.1 Devices using RCS IP video calls

A device/client supporting and configured to use RCS IP video calls shall indicate this according to Table 3.

The device/client shall also include a P-Preferred-Service header with the MMTEL ICSI as per [PRD-IR.94] and include the relevant subclass, i.e.

P-Preferred-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall

The use of the media feature tag *+g.gsma.rcs.ipcall* allows the Service Provider to differentiate between the regular MMTEL (VoLTE/VoHSPA) video call service (without this additional media feature tag) and the RCS IP video call service (with the additional media feature tag) which indicates a strong preference for no breakout (IP end to end).

3.9.4.1.1 IP Video Calls when IP Voice Calls are not supported

If due to configuration (i.e. the values of the PROVIDE RCS IP VOICE CALL and PROVIDE RCS IP VIDEO CALL defined in section A.1.14) a client supports an RCS IP Video Call but does not support user-switch to RCS IP Voice calls, the client shall not accept IP Calls that do not include video media in the SDP offer; however the client shall allow video to be removed from an ongoing RCS IP Video Call if the video is removed by the remote peer. It shall include the *+g.gsma.rcs.ipvideocallonly* feature tag in the Contact and Accept-Contact header fields of the SIP INVITE requests and 200 OK responses that it sends for RCS IP Video Calls.

Similarly if a network element in the path between two clients allows for RCS IP Video Calls and not for RCS IP Voice Calls establishment (e.g. to enforce the interworking agreement for a particular NNI), that network element shall ensure that this restriction is reflected in the exchanged capabilities and include the *+g.gsma.rcs.ipvideocallonly* feature tag in the Contact and Accept-Contact header fields of the SIP INVITE requests and 200 OK responses that are exchanged between the clients for RCS IP Video Calls. The network element shall then also ensure that an RCS IP Call is torn down or rejected if the SDP offer or answer does not include a video media stream.

If a client supporting RCS IP Video Calls receives the *+g.gsma.rcs.ipvideocallonly* feature tag in the Accept-Contact or Contact header fields of respectively the SIP INVITE request or 200 OK response for an RCS IP Video Call, it should not modify the session removing the video stream (i.e. the video media line in the SDP) during an ongoing RCS IP Video Call or not remove the video media line in the SDP answer in case of the recipient. The client supporting RCS IP Video Calls may offer the option to turn the video stream into a uni-directional stream.

3.9.4.2 Flows

3.9.4.2.1 Assumptions

The following sections describe the relevant message flows and reference UX. Please note that the following assumptions have been made:

- For simplicity, the internal mobile network interactions are omitted in the diagrams shown in the following sections.
- For simplicity RTCP exchanges are omitted in the diagrams. They should be executed as described in [PRD-IR.94] and section 2.8
- The terminal comes with a front and rear camera. If one or both are missing, the user should be notified only with the available options.
- The capability exchange was performed already (as described in section 2.6). Both users are thus aware that the other party supports IP Voice and Video calls.

3.9.4.2.2 Direct Launch

3.9.4.2.2.1 Accept as bidirectional

In this scenario no voice call is ongoing between the users and User A decides to initiate a video call with User B. User B accepts the call as a fully bidirectional video call. This results in two bidirectional RTP/RTCP streams, one for the audio and one for the video.

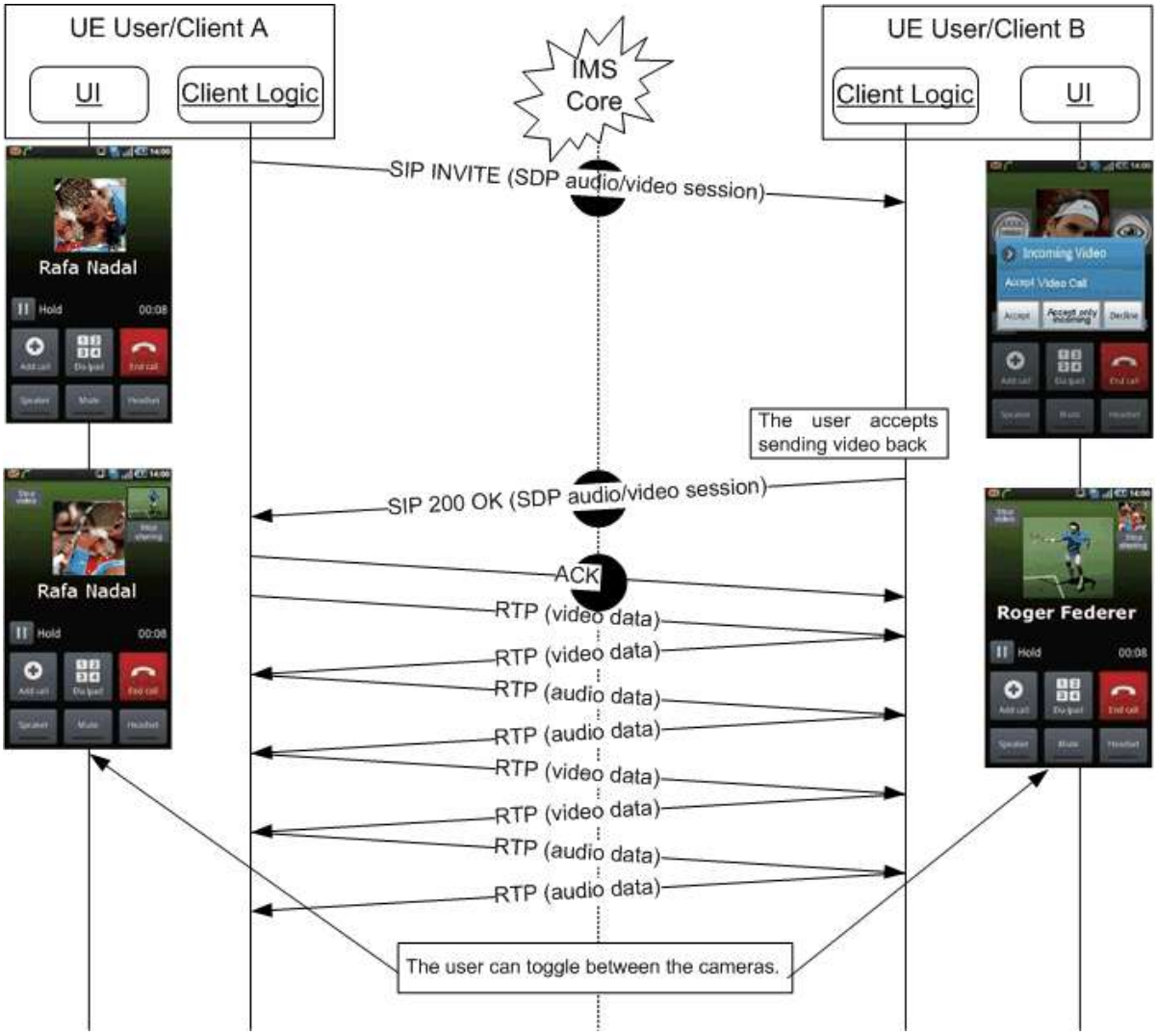


Figure 119: Direct launch of video call - Accept as bidirectional

3.9.4.2.2.2 Accept unidirectional

In this scenario no voice call is ongoing between the users and User A decides to initiate a video call with User B. User B accepts the call, but indicates that they do not want to send video back. This results in two RTP/RTCP streams, one bidirectional for the audio and one unidirectional (from User A to User B) for the video.

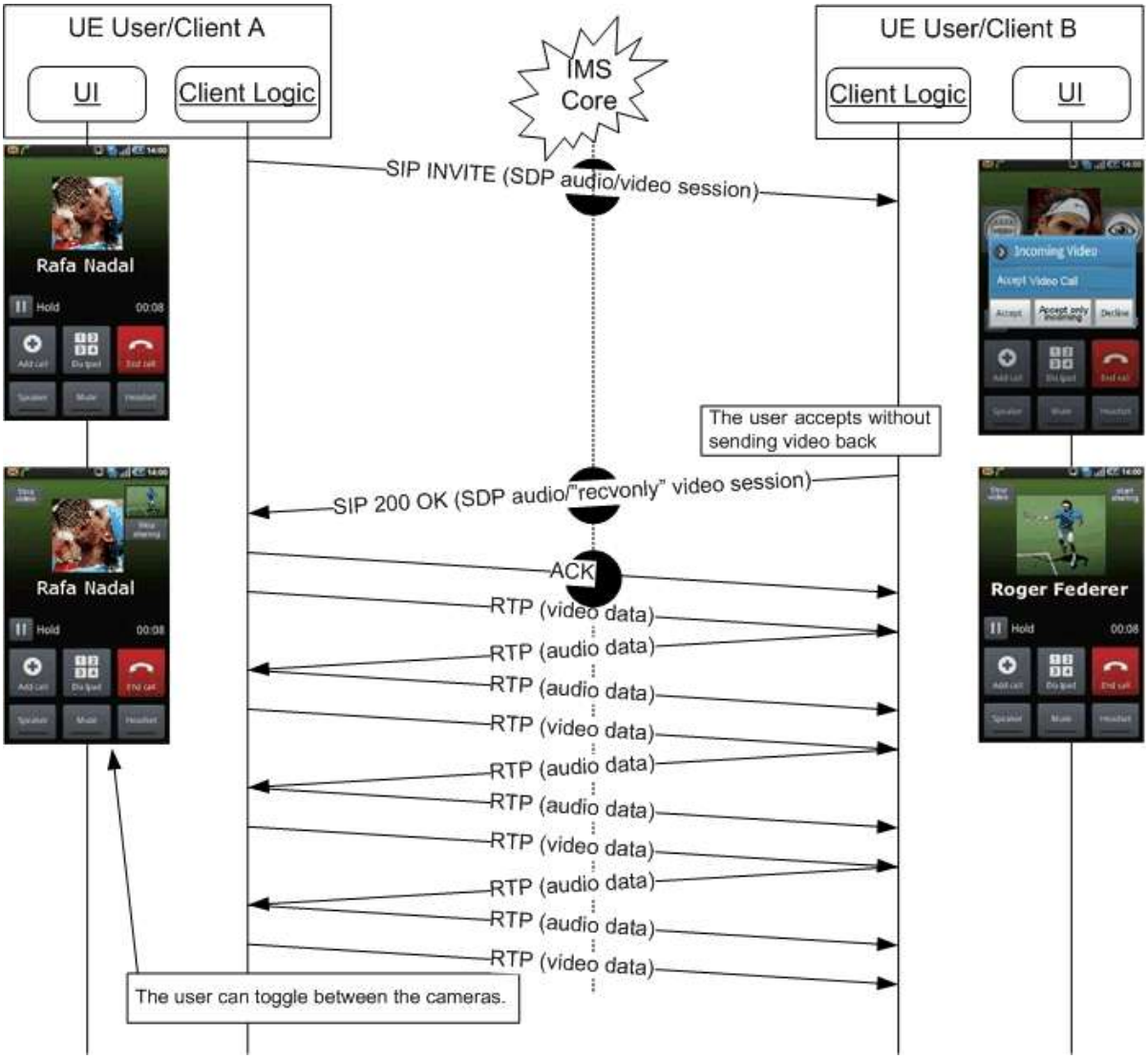


Figure 120: Direct launch of video call - Accept as unidirectional

3.9.4.2.2.3 Decline

In this scenario no voice call is ongoing between the users and User A decides to initiate a video call with User B. User B rejects the call.

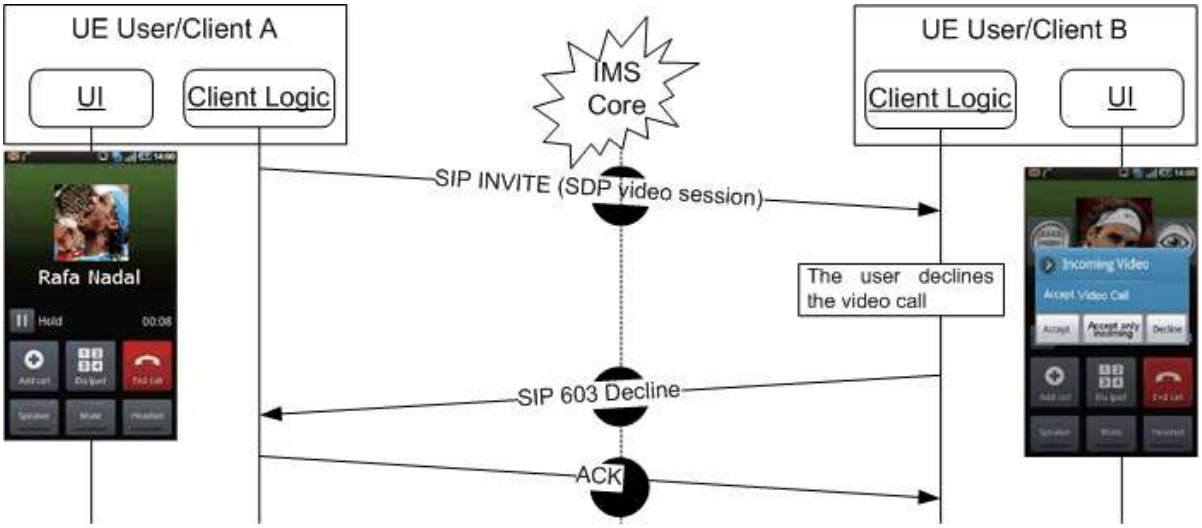


Figure 121: Direct launch of video call – Decline

In this scenario User B's network could also redirect the call to an announcement or voice/video mail system.

3.9.4.2.3 Upgrade from PS Call

3.9.4.2.3.1 Accept

In this scenario a PS voice call is ongoing between the users as specified in section 3.8. As specified in [PRD-IR.94] at the start of this call both terminals have indicated that they are capable of upgrading to a video call and no further capability exchange was done after the call setup indicating that this capability is no longer available.

User A decides to upgrade the ongoing call into a video call. User B accepts the upgrade (and in the illustrated flows decides to send video back). This results in a second RTP/RTCP stream for the video being added to the ongoing call (next to the existing bidirectional audio stream). This video stream can either be bidirectional or unidirectional depending on whether User B accepted to send video back or not. This is similar to the cases illustrated in sections 3.9.4.2.2.1 and 3.9.4.2.2.2.

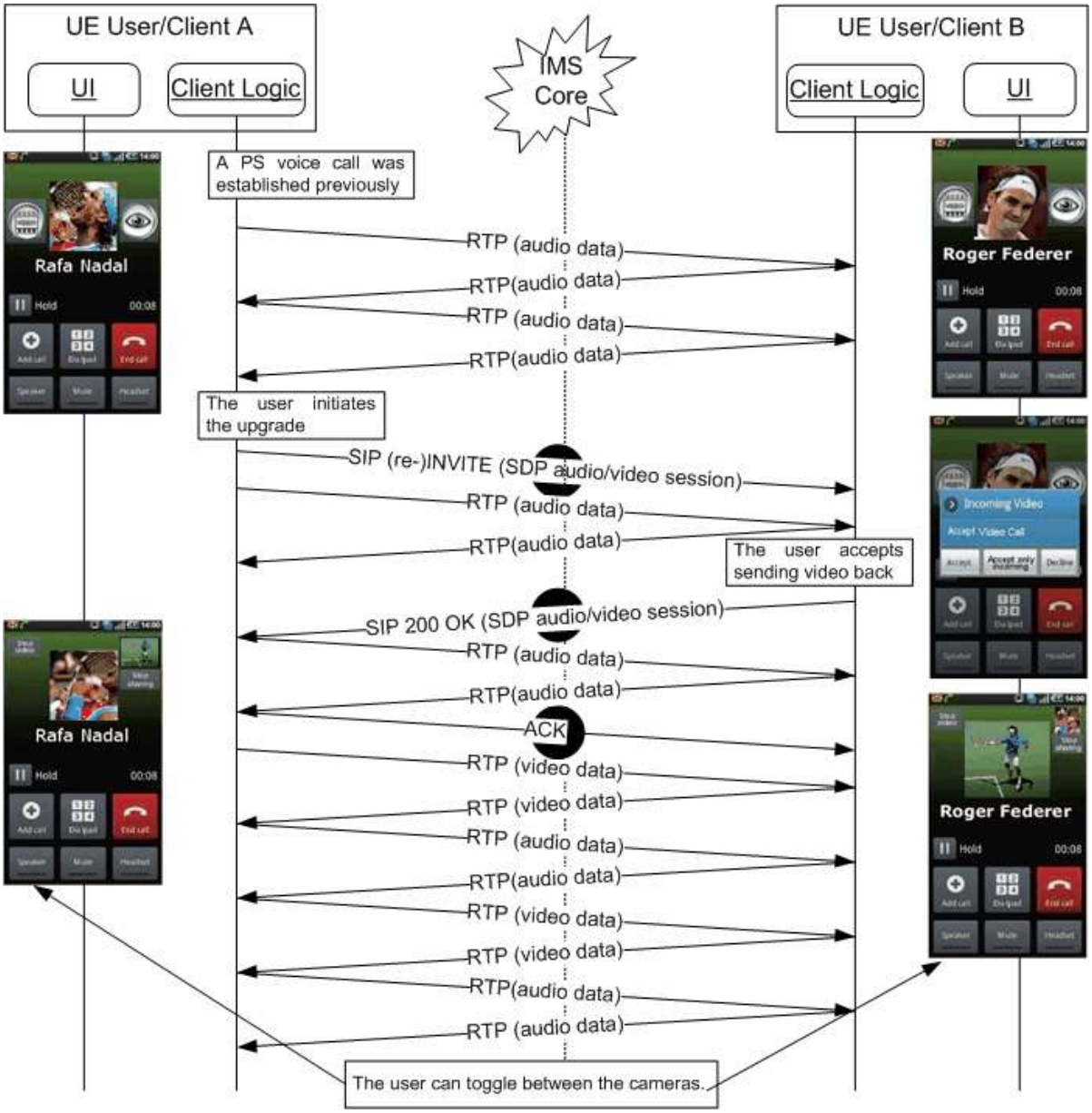


Figure 122: Upgrade PS Voice call to video call

NOTE: in a multidevice scenario the devices from User B that are not involved in the voice call will not be included in this upgrade flow.

3.9.4.2.3.2 Decline

In this scenario a PS voice call is ongoing between the users as specified in section 3.8. As specified in [PRD-IR.94] at the start of this call both terminals have indicated that they are capable of upgrading to a video call and no further capability exchange was done after the call setup indicating that this capability is no longer available.

User A decides to upgrade the ongoing call into a video call. User B declines the upgrade. The voice call continues unaffected.

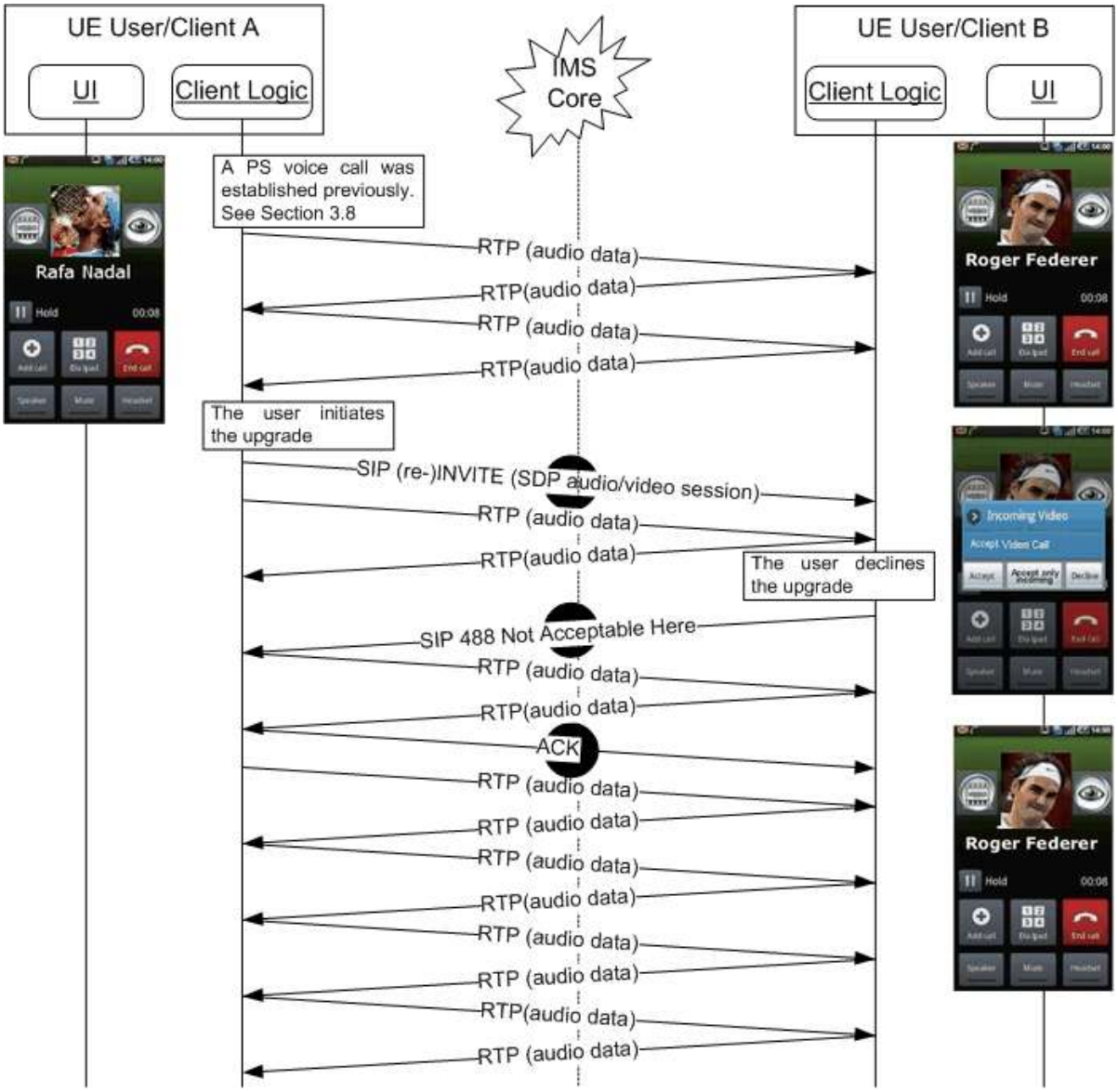


Figure 123: Decline upgrade PS Voice call to video call

3.9.4.2.4 Switch from unidirectional to bidirectional video

In this scenario User A and User B are involved in a video call in which User B is not sending video to User A. Then User B decides to start sending a video stream to User A.

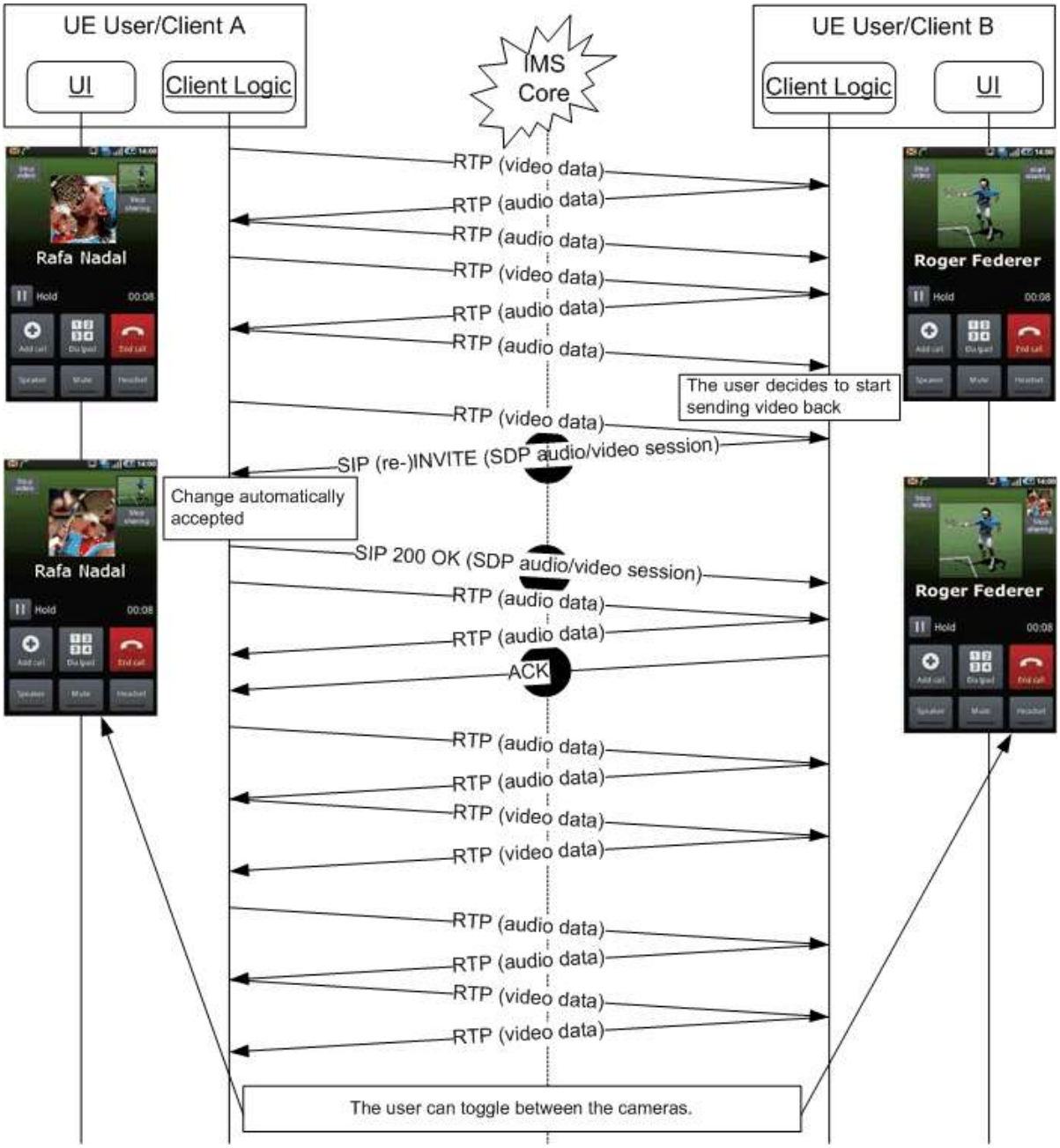


Figure 124: Change from unidirectional video call to bidirectional video call

3.9.4.2.5 Switch from bidirectional to unidirectional video

In this scenario User A and User B are involved in a video call in which both users are sending video to each other. Then User B decides to stop sending a video stream to User A.

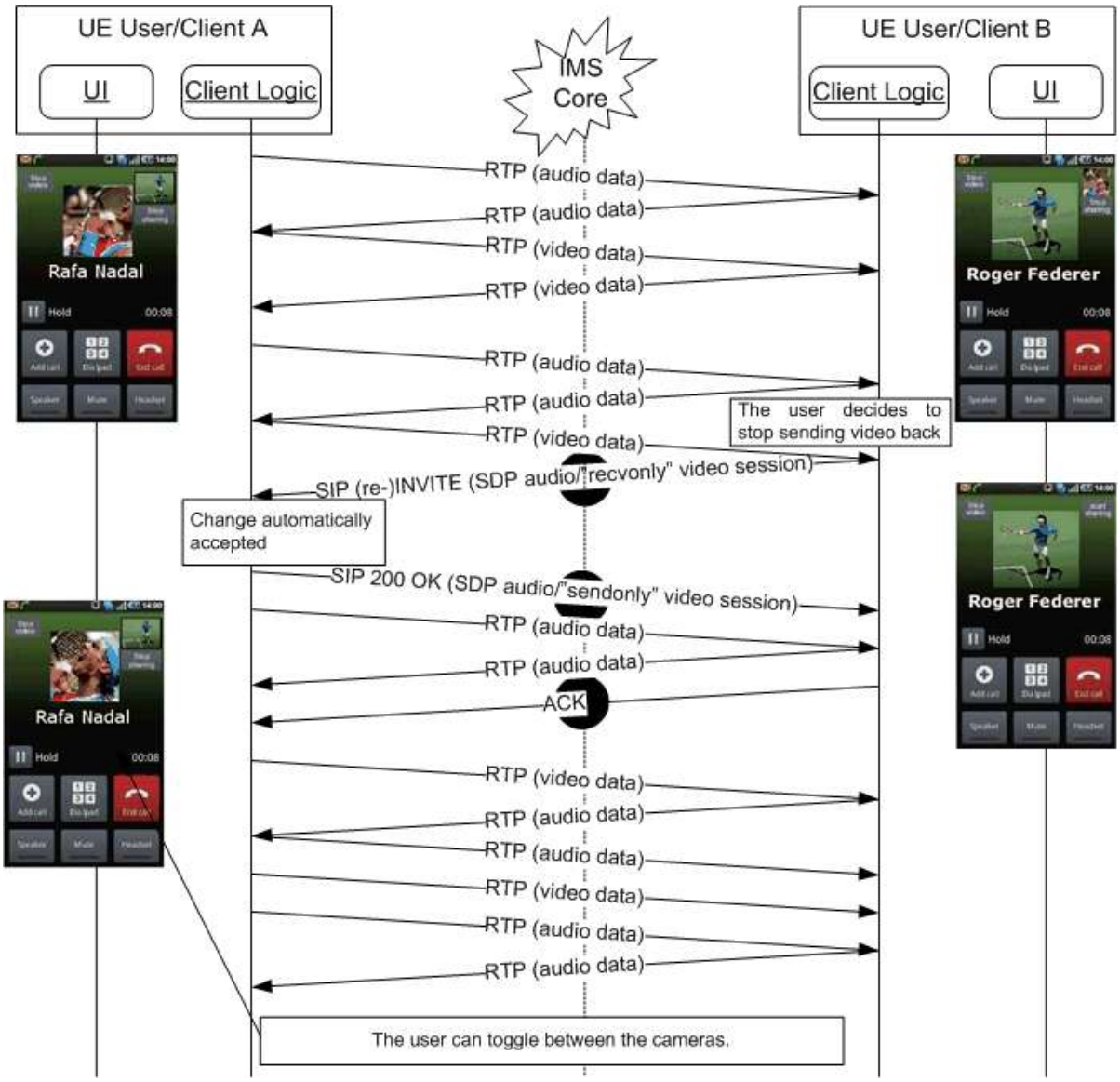


Figure 125: Change from bidirectional video call to unidirectional video call

3.9.4.2.6 Video call termination

In this scenario User A and User B are involved in a video call with each other and User A decides to terminate the call.

NOTE1: in this scenario User A is not necessarily the user that started the call.

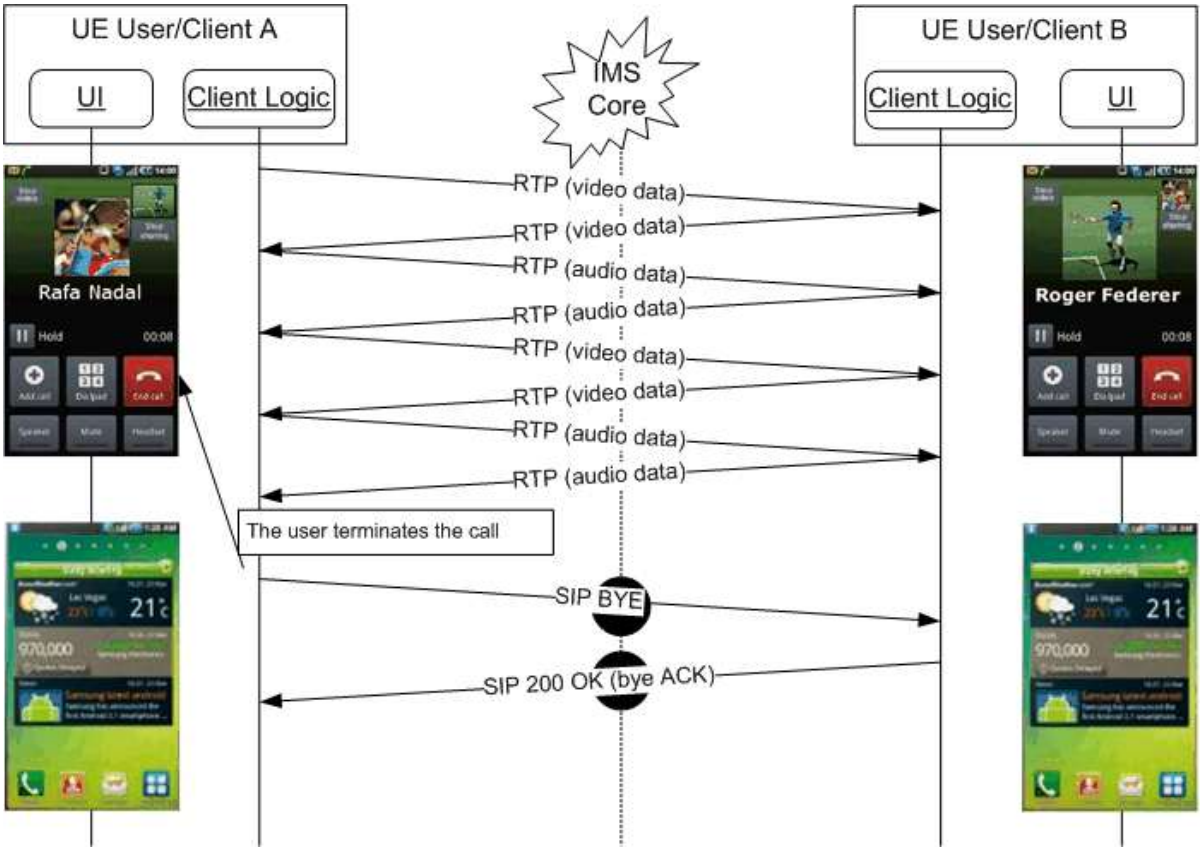


Figure 126: Video call termination

NOTE2: As this terminates the communication between User A and B, there is no need to do a capability exchange to verify whether the termination was or was not voluntary in contrast to the situation for Video Share described in section 3.6.4.

3.9.4.2.7 Replace CS Voice Call with an IP Video Call

3.9.4.2.7.1 Accept

In this scenario a CS voice call is ongoing between the users.

User A decides to upgrade the ongoing call into a video call. The CS call is torn down by User A's device before initiating the IP Video Call. From an IMS network perspective the flow for the IP Video Call is identical to direct launch, as described in section 3.9.4.2.2. The video stream can either be bi-directional or uni-directional depending on whether User B accepted to send video back or not. This is similar to the cases illustrated in sections 3.9.4.2.2.1 and 3.9.4.2.2.2.

As an alternative to the flow in Figure 127, before tearing down the CS call, by using the configuration parameter *RCS IP VIDEO CALL UPGRADE ATTEMPT EARLY* User A's client could attempt right away to set up the RCS IP Video Call once User A decides to do the upgrade, but if a service interaction issue causes the attempt to fail (e.g. User B does not have Call Waiting so the incoming RCS IP Video Call attempt fails with a busy signal), then User A's mobile could proceed to tear down the CS call and then initiate the Video Call attempt.

A second alternative to the flow in Figure 127, is to, by using the configuration parameter *RCS IP VIDEO CALL UPGRADE ALLOWED ON CAPABILITY ERROR*, let the user attempt

an RCS IP Video Call even if a 480/408 error response is returned and no knowledge of service capabilities is available for that user.

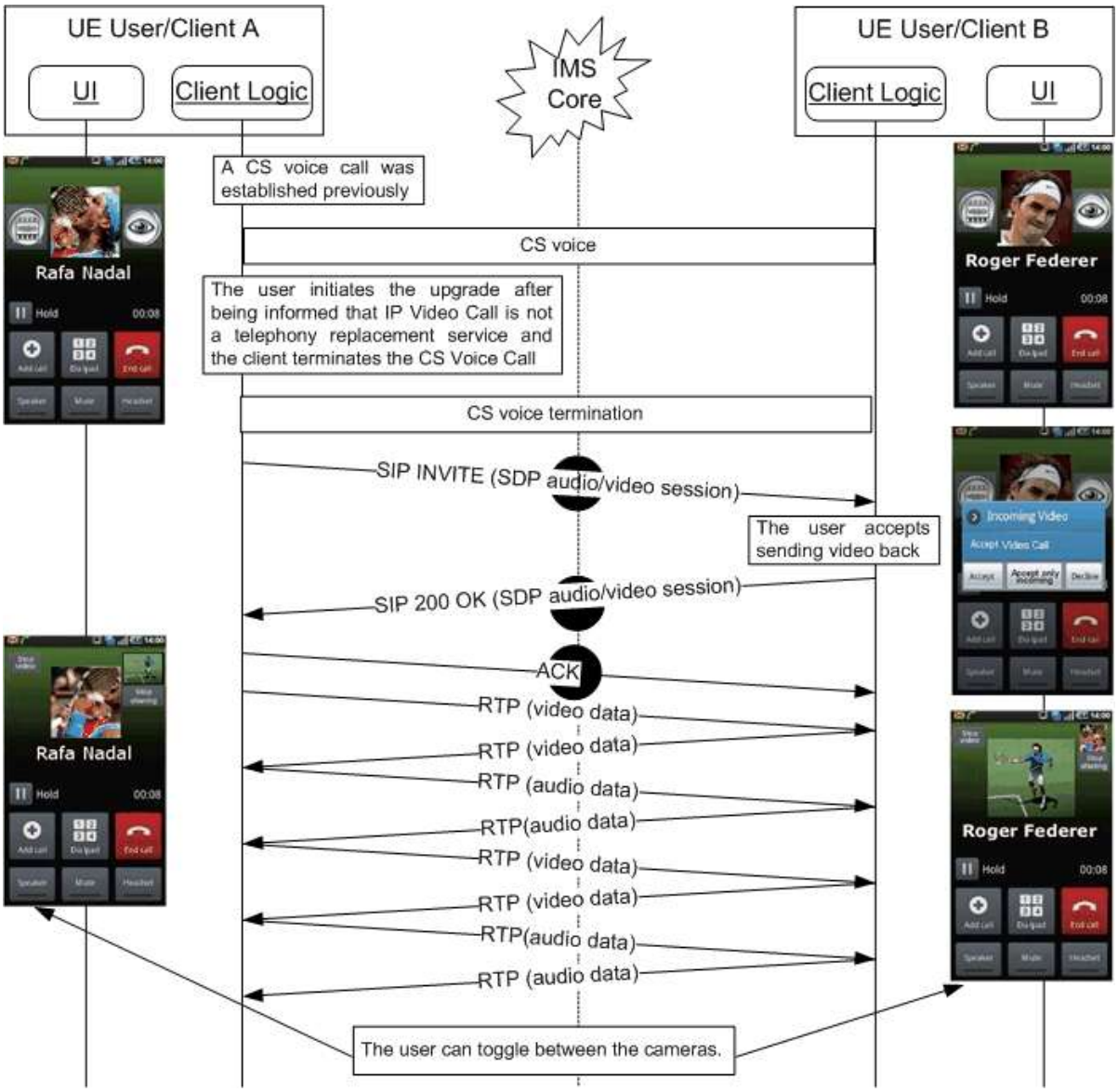


Figure 127 : Replace CS Voice call with a RCS IP Video Call

NOTE: in a multidevice scenario the devices from User B that are not involved in the voice call will be included in this flow.

3.9.4.2.7.2 Decline

In this scenario a CS voice call is ongoing between the users.

User A decides to upgrade the ongoing call into a video call. The CS call is torn down by User A's device before initiating the IP Video Call. User B declines the upgrade. At this point User A should redial User B via CS voice.

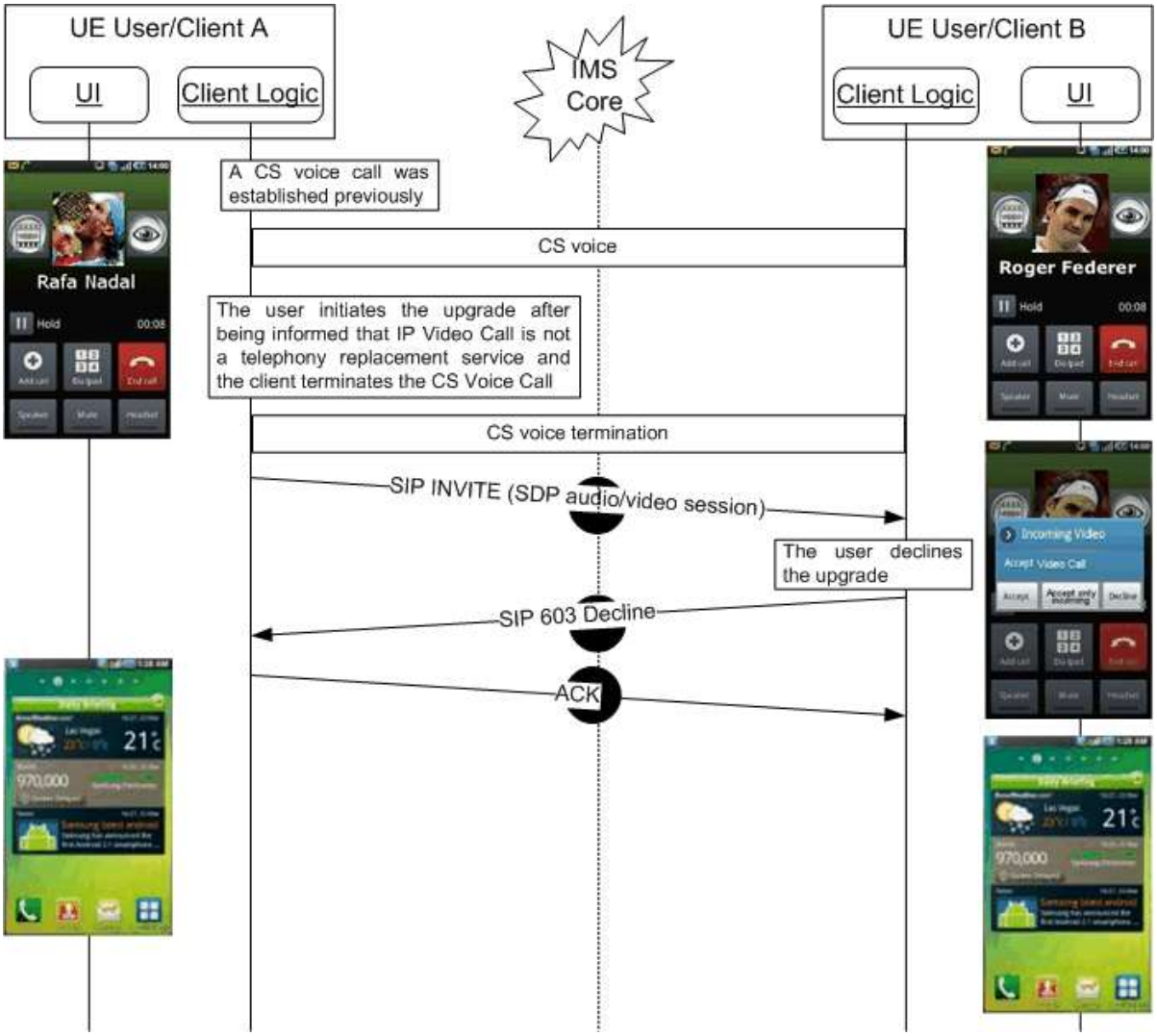


Figure 128: Decline replacement of CS Voice call with an RCS IP Video Call

3.9.4.2.7.3 Incoming CS Voice call when already in an IP Video Call

In this scenario an RCS IP Video Call is ongoing between two users.

User A receives an incoming CS Voice call from User C. User A shall receive the announcement of the incoming call. User A shall be able to:

1. Reject the incoming CS Voice call from User C (and thus stay in the same IP Video Call with User B as long as data connectivity was not lost);
2. Accept the incoming CS Voice call from User C, and consequently tear down the IP Video Call with User B;
3. Put the IP Video Call with User B on hold (as long as data connectivity was not lost) and answer the incoming CS Voice call from User C.

NOTE1: If the device is using LTE and the CS Fallback network is 2G, the data connection is suspended and resumed after the CS call, so the IP Video Call may or may not still be there.

NOTE2: when a Service Provider's deployment allows directing incoming RCS IP Video Calls to devices that are in a CS voice call already, equivalent options will be available to the user unless the IP Video Call is originated by the same user as the conversation partner in the CS voice call. In that case, the

incoming invitation for an IP Video Call should be presented to the user as the possibility to accept or deny the upgrade of the voice call to an IP Video call which also means that option 3 would not be available.

3.9.5 NNI and IOT considerations

The NNI interfaces for content sharing services shall behave according to the procedures described in section 2.12 and the documents it refers to.

3.9.6 Implementation guidelines and examples

From the UX perspective, there are three possible entry points to these services:

1. Address book/Call-log: A video call can be initiated with any registered contact providing the right capabilities are in place – contact oriented initiation.

Figure 129: User experience when starting from address book

2. Chat window: From the Chat (one-to-one Chat only) window a video call can be initiated using the relevant menu item. The experience is identical to the address book/call-log. The capability query is initiated when the user opens up the menu in which the available communication options are offered

Figure 130: User experience when starting from chat

3. Call screen: an ongoing voice call can be upgraded to a video call.

Figure 131: User experience when starting from call screen

Regardless of whether it is an upgrade scenario or a direct call, the receiver will always get 3 options on an incoming video call:

1. Accept
2. Accept only to receive video
3. Decline

Figure 132: Video call receiver user experience direct video call

Figure 133: Video call receiver user experience – upgrade from voice call

3.9.6.1 Multidevice handling

When receiving an incoming IP Voice Call with video capabilities indicated as specified in [PRD-IR.94], it is recommended to have the recipient's devices supporting the IP Video Call display a video upgrade indication while it is alerting in order to draw the user's attention to the fact that answering at that device will allow the possibility to upgrade to a video call during the voice call.

3.10 Geolocation services

3.10.1 Feature description

Geolocation services comprise the following 2 features:

1. The "Geolocation PUSH" service that allows an RCS user to push location information (that can be the user location or the location of a suggested meeting point) to another RCS user
2. The "Geolocation PULL" service that allows an RCS user to retrieve the location information about another RCS user

It should be noted that similar services can be provided through the SPI with geolocation presence information (see section 3.7).

Their introduction in RCS is justified by the fact that an RCS user can have an interest to share geolocation information when SPI geolocation information cannot be used:

- Because SPI service is not offered by the Service Provider (if the 2 users belong to the same Service Provider) or one of the 2 Service Providers (if the 2 users do not belong to the same Service Provider)
- Because SPI is offered by the Service Provider (or the 2 Service Providers if the 2 users do not belong to the same Service Provider), but the 2 users do not want to share social information

3.10.1.1 Geolocation PUSH feature

Locations can be selected by the sender as follows:

- push current location
- push pre-defined location (e.g., the home address, a tool which permits a user to select from 'favourite locations' may be provided)
- push a location that is selected on a map

The user can also choose to put additional text information about the location

The full user experience is possible only between two RCS 5.1/5.2 users. This is ensured by the RCS Service Discovery scheme.

3.10.1.2 Geolocation PULL feature

This feature is used by an RCS user, the origin RCS user, to retrieve the location information on any other RCS user – i.e. not limiting to users that share SPI with the RCS user

Behaviour at the origin RCS user side:

- When successful, the RCS user is informed with the result: geolocation coordinates (x, y).
- The user can then choose to store the information in the address book or/and show the information on a map

Behaviour at the target RCS user side

- The target user is informed that another RCS user is requesting to retrieve their geolocation
- The target user either authorizes (ALLOW) or refuses (DENY) to share their geolocation
- If the target authorizes (ALLOW) sharing their location, the location is retrieved automatically by the client/device accessing the Location Based Services (LBS) infrastructure in the network.

Multi device handling for the Geolocation PULL feature:

- The primary device will be the default recipient of the authorization request. If the user replies 'ALLOW', this primary device will provide the user location information

3.10.2 Interaction with other RCS features

3.10.2.1 Geolocation PULL service

Interaction with RCS chat and voice/video call: the feature can be activated in the context of an established voice/video call (single point or multipoint) or in the context of an established RCS chat.

3.10.2.2 Geolocation PUSH service

Interaction with RCS chat and voice/video call: the feature can be activated in the context of an established voice/video call (single point or multipoint) or in the context of an established RCS chat.

The Geolocation PUSH service can also be used in the context of a 1-to-1 Chat, a Group Chat or a Call to deliver the "Show on a Map" functionality.

3.10.3 High Level Requirements

3.10.3.1 Geolocation PUSH

3-10-1 Geolocation information should be made available to any user (notwithstanding whether at home-PLMN or roaming in visiting-PLMN)

3-10-2 Shall be deployed as point to point service between 2 RCS users having the capability

3-10-3 An RCS user shall have the possibility to communicate geolocation information to a contact that has Geolocation PUSH capability

3-10-4 The service can be accessed from the address book or share menu

3-10-5 The service can be accessed also within a call, a chat or a Group Chat

3-10-6 Geolocation information shall consist of:

- Free text entered by the RCS user (optional)

- coordinates (x,y) (mandatory)

3-10-7 Coordinates can be obtained Manually

- The user referring to a predefined stored location
- Or the user picks the location point on a map.

3-10-8 Coordinates can be obtained Automatically (via one of the localisation methods available in the device and the network)

3-10-9 The user can choose the precision of the location that they want to communicate a Street, City or Country for example

3-10-10 If authorized by the Service Provider (GEOLOCATION VALIDITY parameter in section A.1.7.2), the user has the option to enter a validity time for the geolocation information

3.10.3.2 Geolocation PULL

3-10-11 Geolocation information should be made available to any user (notwithstanding whether at home-PLMN or roaming in visiting-PLMN)

3-10-12 Shall be deployed as point to point service between 2 RCS users having the capability

3-10-13 An RCS user (Emitter side) shall have the possibility to retrieve geolocation information from a contact that has Geolocation PULL capability

3-10-14 The service can be accessed through the address book

3-10-15 The service can be accessed also within a call, a Chat or a Group Chat

3-10-16 The contact (Receiving side) shall have the possibility to accept or to deny the request

3-10-17 There is an expiration period for the authorization granted by the target subscriber. The authorization is on per application (RCS) and per requesting subscriber basis.

3-10-18 The subscriber shall be able to STOP the authorization at any time before the expiration period ends

3-10-19 In case of DENY or STOP, the user shall have the possibility to REVOKE the originator of the Geolocation PULL request. In this case, the originator is put in a Geolocation PULL black list

3-10-20 If the Receiving side accepts the demand, geolocation information provided by the LBS infrastructure in the network consists of: coordinates (x,y)

3-10-21 If authorized by the Service Provider (GEOLOCATION VALIDITY parameter in section A.1.7.2), the user shall have the option to enter a validity time for the geolocation information when the target user is replying to allow PULL operation

3.10.3.3 Show on a Map

3-10-22 It shall be possible to show the locations of the participants in a 1-to-1 or Group Chat or a call together on a map

3.10.4 Technical Realization

3.10.4.1 Geolocation PUSH service

The RCS File Transfer service (see previous sections 3.5.4.1 and 3.5.4.3 to 3.5.4.7) is used to convey the geolocation information during a voice or video call (assuming the person the user wants to send his location to is the one in the call).

When there is no such communication context or there is an already established Chat session (1-to-1 Chat or Group Chat), the geolocation information shall be sent directly as a message in a Chat session provided the intended recipient (for a 1-to-1 Chat) or the controlling function (for a Group Chat) supports Geolocation Push. In both cases the same

format shall be used which is described in section 3.10.4.3. As for File Transfer via HTTP, message revocation procedures as described in section 3.3.4.1.10, do not apply for 1-to-1 Chat messages carrying geolocation information.

3.10.4.1.1 Geolocation PUSH during a voice or video call

3.10.4.1.1.1 Requester side

The Geolocation PUSH service is proposed to the user if the Service Discovery Process has determined that the target RCS user has the Geolocation PUSH service available. See chapter 2.6 and chapter 2.6.4.1 for Service Discovery. An RCS user having the RCS Geolocation PUSH capability must have also the RCS File Transfer capability

If the user has chosen to provide his/her location through automatic localization:

- The RCS user's device is to use OMA SUPL (user plane) technology as the preferred mechanism for obtaining the geolocation (SET initiated primitive);
- If SUPL is not supported by a Service Provider, the RCS user' s device is free to use other locating method(s) rendering the highest possible precision in obtaining geolocation information

The geolocation application interfaces with the RCS File Transfer enabler

Next to the standard SDP parameters for RCS File Transfer, the application of the sender has to set the parameter *type* of the *file-selector* attribute to *application/vnd.gsma.rcspushlocation+xml*

The SIP File Transfer uses a specific IARI that allows routing the primitive to the geolocation application in the target device B. The file transfer name has no meaning in this case.

The file type is *application/vnd.gsma.rcspushlocation+xml*. See the section 3.10.4.3 for more details.

3.10.4.1.2 Receiving side

The RCS File Transfer request is routed to the RCS geolocation application (internal routing based on the IARI).

On the receiving side the File Transfer invitation complies with the acceptance rules of RCS File Transfer.

If the transfer is successful, the application triggers the user in a pop up menu to handle the location information.

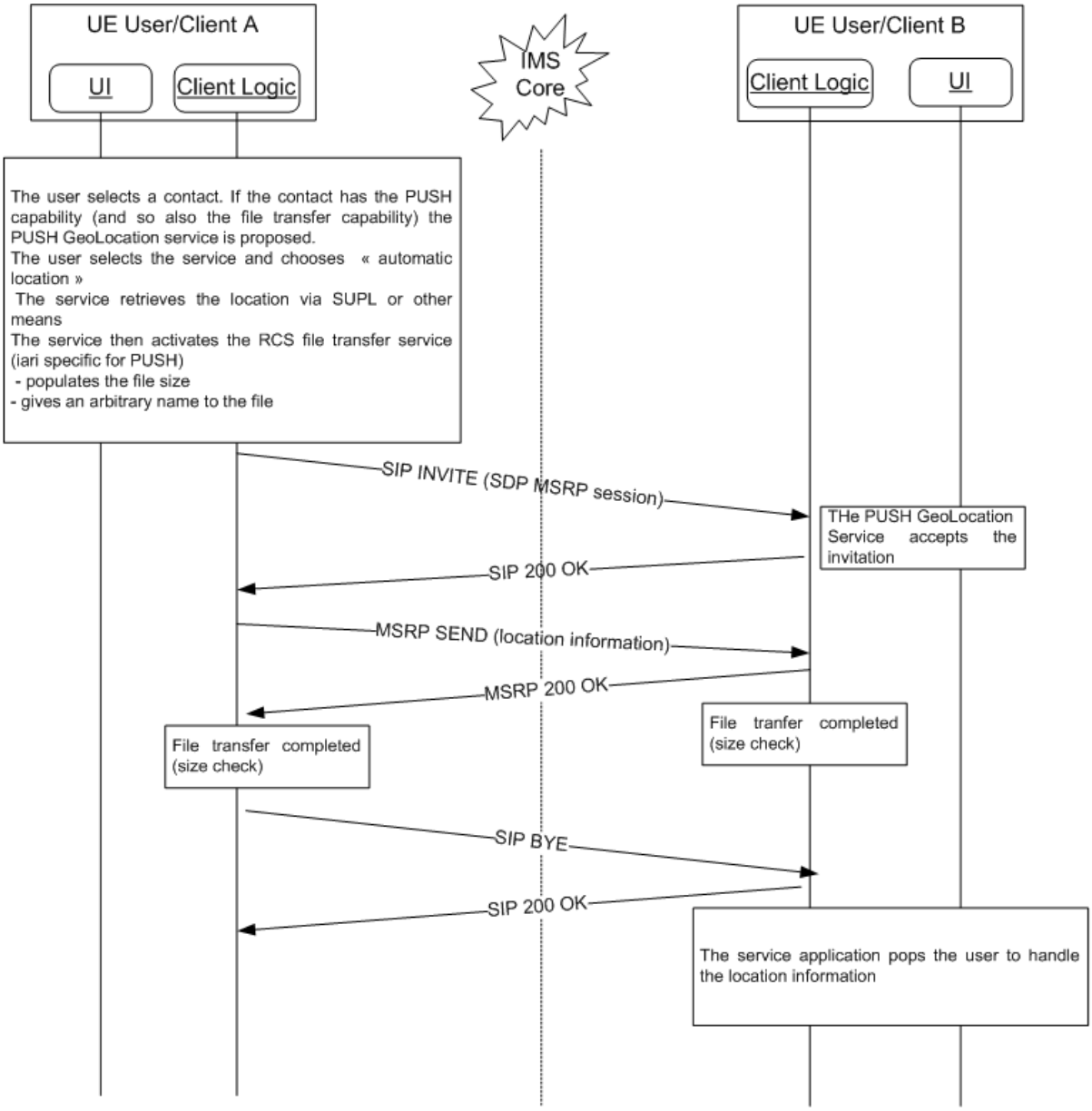


Figure 134: Push of geolocation information during a voice or video call using RCS File Transfer

3.10.4.1.3 Geolocation Push Outside of a voice or video call

When not in a call the Geolocation PUSH service shall transfer the location information between users by sending the Geolocation Content directly in a Chat Message. That allows potentially reusing an already established 1-to-1 or Group Chat session for Geolocation PUSH.

In an active Chat session the transfer shall be possible if

- the Geolocation PUSH content type was included in the a=accept-wrapped-types attribute of the SDP received during the setup of the Chat session and
  - In case of a 1-to-1 session, the contact supports Geolocation PUSH (i.e. the corresponding capability was discovered or was cached)
  - In case of a Group Chat, the Contact header received during the setup of the Group Chat included the Geolocation PUSH IARI tag defined in section 2.6.1.1.2.

When these conditions are fulfilled a client can transfer the geolocation information in a CPIM wrapper that is transferred using an MSRP SEND request..

3.10.4.1.3.1 1-to-1 Exchange of Geolocation PUSH outside of a voice call

In case a new 1-to-1 session needs to be established when the user wants to transfer geolocation information to a contact that has the Geolocation PUSH capability, the sending client shall generate a SIP INVITE request for a 1-to-1 Chat session and include an additional *Accept-Contact* header field in the SIP INVITE request carrying the Geolocation PUSH IARI along with the *require* and *explicit* parameters. This will ensure that the request is routed only to Geolocation PUSH capable devices which shall handle the acceptance of the received SIP INVITE request in the same manner as that of a regular Chat INVITE request (i.e. controlled through the IM SESSION START and IM SESSION AUTO ACCEPT configuration parameters). The Geolocation PUSH XML message body itself shall then be sent as first message in the Chat following the configuration parameter FIRST MSG IN INVITE defined in Table 85 in section A.1.3.3.

If there is an active 1-to-1 Chat session with a Geolocation PUSH capable contact, but the *a=accept-wrapped-types* SDP attribute received during the setup of that Chat session did not include the *application/vnd.gsma.rcspushlocation+xml* MIME content type, Geolocation PUSH to that contact will not be available.

3.10.4.1.3.2 Multiparty Exchange of Geolocation PUSH

During Group Chats, the capability to use Geolocation PUSH depends on the controlling function. A Geolocation PUSH capable controlling function shall enable Geolocation PUSH by including the *application/vnd.gsma.rcspushlocation+xml* MIME content type in the *a=accept-wrapped-types* SDP attribute that it provides during the setup of the Group Chat Session and include the Geolocation PUSH IARI tag defined in section 2.6.1.1.2 in the Contact headers that it includes in the SIP INVITE requests and 200 OK responses for the setup of the Group Chat. A Geolocation PUSH capable controlling function shall not distribute Geolocation PUSH information to the participants in the chat that are not Geolocation PUSH capable. A client on which Geolocation PUSH was enabled shall during the setup of the Group Chat indicate to the controlling function that it supports Geolocation PUSH by including the *application/vnd.gsma.rcspushlocation+xml* MIME content type in the a=accept-wrapped-types SDP attribute and the Geolocation PUSH IARI tag defined in section 2.6.1.1.2 in the Contact headers of the SIP INVITE requests and 200 OK responses that it generates. When during a Group Chat the *a=accept-wrapped-types* SDP attribute received by a client or conference focus did not include the *application/vnd.gsma.rcspushlocation+xml* MIME content type or the Geolocation PUSH IARI tag was not provided in the received Contact header, Geolocation PUSH shall not be available for the Group Chat in which the client participates and for a specific client in the Group Chat respectively.

When the users wants to send the Geolocation information to the participants of an existing Group Chat that is idle, a client that is configured to support Geolocation PUSH shall first restart the chat and then send the file in the chat.

When the user wants to send geolocation information to multiple contacts outside of the context of an existing Group Chat, a client that is configured to support Geolocation PUSH shall first start a new Group Chat with the selected contacts and send the Geolocation XML body as first message in the chat.

3.10.4.2 Geolocation PULL service

There are 2 possible technical solutions to offer the service:

1. Based on a Location API gateway and a LBS network infrastructure
2. Based on “fetch” file transfer facility.

An RCS service provider may offer one or both of these technical solutions.

Different Services Identifiers in the capability exchange (see section 2.6 and chapter 2.6.4.1) allow determining what Geolocation technical solution(s) a target RCS contact supports.

The Geolocation PULL service is proposed to the user if the capability exchange has determined that the target has the service available (i.e. at least one of the technical solutions is available) and the service operator has authorized the service to be used with at least one of the available technical solutions. The parameter PROVIDE GEOLOC PULL (See Annex A, Table 90) allows the Service Provider to indicate his preferred policy.

If both technical solutions are supported by a target RCS contact, it is up to the Geolocation application on the requester’s device to select the appropriate technical solution based on the chosen operator policy.

3.10.4.2.1 Technical solution based on Geolocation API gateway and LBS infrastructure

This service is realised using the OMA NetAPI_TerminalLocation API [Location_API] and its complement GCOP (GSMA Canadian OneAPI Pilot) Privacy_Service API [PRIVACY-API].

3.10.4.2.1.1 Requester side

The Geolocation PULL service, if available for the user, is proposed

- When the service provider has chosen to restrict the service to RCS target users (see parameter *GEOLOCATION PULL OPEN* in Table 90 in section A.1.7.2) and the Service Discovery Process has determined that the target has the service available. See chapter 2.6 and chapter 2.6.4.1 for Service Discovery
- Independent of whether the target contact is an RCS user if the service provider has chosen to open the service to enable Geolocation PULL retrieval of non RCS users (see parameter *GEOLOCATION PULL OPEN*)

The geolocation application interfaces the 2 APIs mentioned in section 3.10.4.2 to obtain authorisation and retrieve location

3.10.4.2.1.2 Receiving side

Authorization request /answer: The authorization request is received by the device through a standard user SMS:

```
The <User_x> wants to use your location. Reply ALLOW or DENY. To cancel all location authorizations, reply STOP.
```

The target user replies in a MO-SMS (Mobile Originated SMS) message back to the OneAPI system

If the user has given their authorization, the OneAPI engages its network enabler (via OMA Mobile Location Protocol, MLP) to query the location of the target mobile from the LBS infrastructure (i.e. Gateway Mobile Location Centre (GMLC) and Serving Mobile Location Centre (SMLC)). This is to be a network initiated location query (either Control Plane or SUPL) to the target mobile.

3.10.4.2.2 Technical solution based on file transfer

This second solution is a technical alternative that doesn’t need an underlying LBS infrastructure.

The solution is based on an end to end “pull” CPM/SIMPLE IM file transfer.

The format of the file is identical to the format used for the Geolocation PUSH service (see section 3.10.4.3 for the definition of the format).

In a multi-device environment, the file transfer request must be routed to the mobile device. For that purpose, at IMS registration phase, a Broadband device must not register the IARI associated with the Geolocation PULL Service based on File Transfer (see section 2.6.1.1.2).

3.10.4.2.2.1 Requester side

The Geolocation application builds an OMA SIMPLE IM or OMA CPM (depending on the messaging technology used by the RCS service provider) File Transfer session that includes the SDP attribute a=recvonly in the SIP INVITE request. The requester indicates that he/she wants to receive geolocation information by setting the *type* selector of the *file-selector* attribute (defined in [RFC5547]) to *application/vnd.gsma.rcspushlocation+xml*.

The inclusion of the dedicated IARI value for Geolocation PULL (defined in Table 28) in the *Contact* header field and along with *require* and *explicit* tags in a dedicated *Accept-Contact* header field allows identifying the request as a Geolocation PULL request at the receiver side.

The behaviour of the application depends then on the response to this SIP INVITE request for File Transfer:

- A SIP 200 OK Response: The operation is authorized by the target user. The MSRP session associated with the File Transfer exchange allows the application to retrieve the location information
- 603 Decline: The operation is not authorized by the target user. Reason can be:
  - The target user has explicitly not authorized the operation
  - The requester is blocked by the target user

  In this case, the requester is not allowed to send a similar Location Pull operation to this contact for a duration that is controlled by the service provider (GEOLOCATION PULL BLOCK TIMER defined in Annex A in Table 90)
- A Time out : the operation is not successful, there is no restriction on the application for resending a similar request towards the target user

3.10.4.2.2.2 Receiving side

On receiving a Geolocation PULL in a File Transfer request, the following steps are processed:

- The Geolocation PULL request has to be explicitly authorized by the receiver.
- When authorizing the request the user must provide the following:
  - The accuracy of the location they want to provide to this requester.
  - Optionally a validity time: This is the time the user estimates his current location to remain stable. The value is also controlled by the Service Provider through the GEOLOCATION VALIDITY parameter (see Table 90 in section A.1.7.2)
  - Optionally an authorization validity time (during this time, any other requests from the sender are automatically accepted by the application without consulting the end user for authorization). When not provided, by default the authorisation is granted for one request only. The authorization timer is a client internal timer that is not visible on the UNI interface
- If the request is authorized (either explicitly by the user or automatically when the authorization validity time has not elapsed), the Geolocation PULL application fetches

the location, establishes the File Transfer Session by returning a 200 OK response and uses that to return a file indicating the user's location with indicated accuracy

- If the authorization is denied by the user (either because the requesting user is in the local black list, or the user explicitly rejects the Geolocation PULL request), a SIP *603 Decline* response is returned to the requester
- If the authorization is pending (i.e. the user has not answered the authorization request before the SIP INVITE transaction timer has elapsed), no response is sent back. This will result in a timeout of the SIP INVITE transaction at the requester's side

3.10.4.2.2.3 Flows

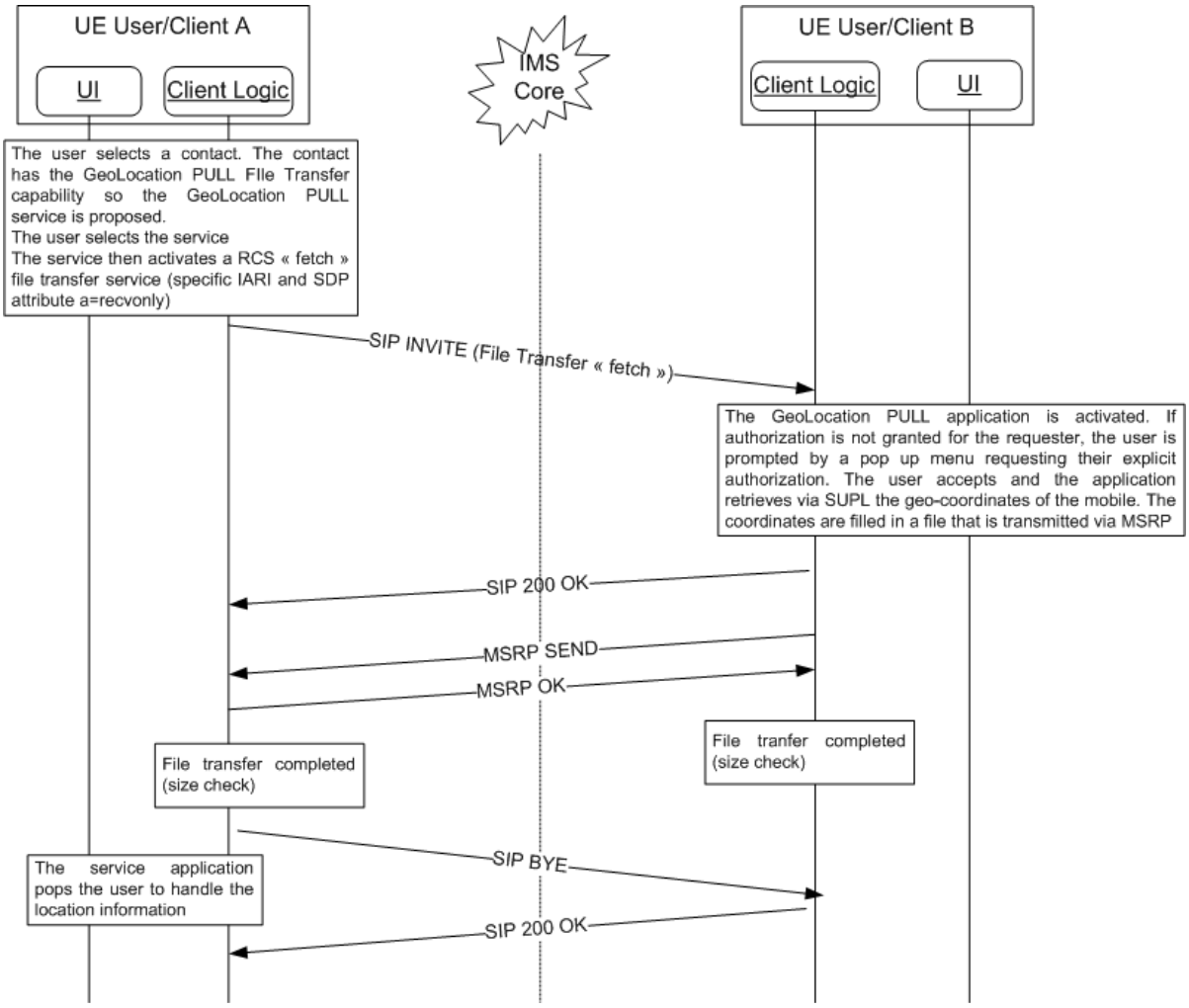


Figure 135: Pull of geolocation information using File Transfer. Success case

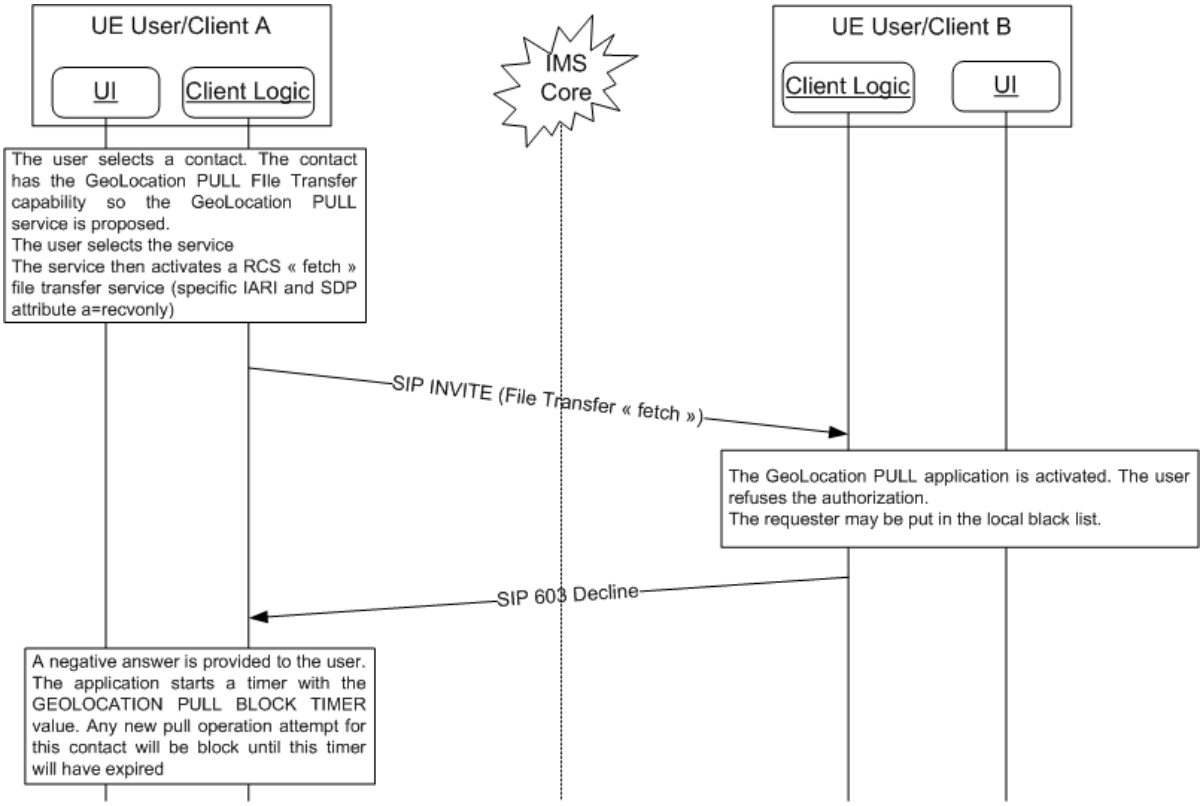


Figure 136: Pull of geolocation information using File Transfer. The target user refuses to give their authorization for the operation

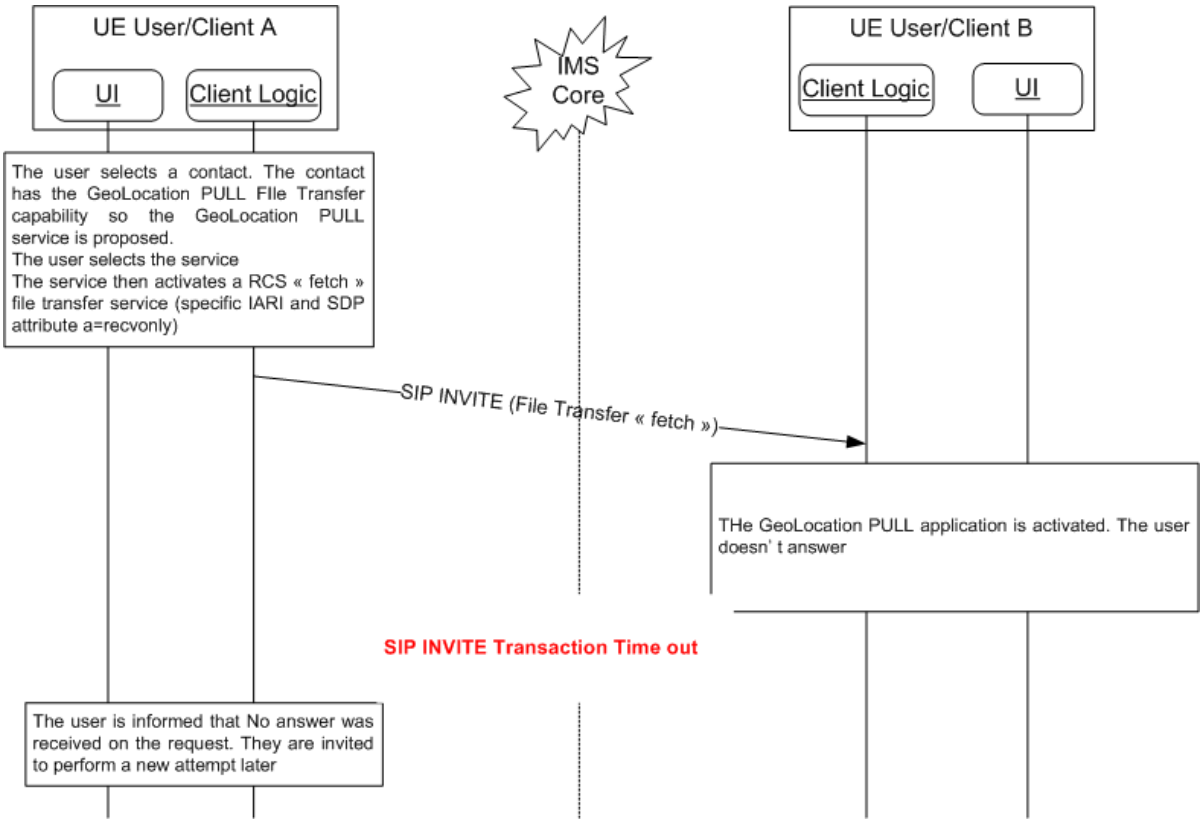


Figure 137: Pull of geolocation information using File Transfer. The target user doesn’t answer

3.10.4.3 Location Information format

3.10.4.3.1 General

The format of the information re-uses the general structure of the RCS XML Presence data. It uses a subset of RCS SPI data definition adapted to RCS Location information

The following XML schema is defined:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:gsma:params:xml:ns:rcs:rcs:geolocation"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns="urn:gsma:params:xml:ns:rcs:rcs:geolocation"
        elementFormDefault="qualified"
        attributeFormDefault="unqualified">
    <xs:element name="rcsenvelope">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="rcspushlocation">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:any namespace="##other" processContents="lax"
                            minOccurs="0" maxOccurs="unbounded"/>
                            <xs:element name="timestamp">
                                <xs:simpleType>
                                    <xs:restriction base="xs:dateTime"/>
                                </xs:simpleType>
                            </xs:element>
                        </xs:sequence>
                        <xs:attribute name="id" type="xs:ID" use="required"/>
                        <xs:attribute name="label" type="xs:string" use="optional"/>
                    </xs:complexType>
                </xs:element>
                <xs:any namespace="##other" processContents="lax" minOccurs="0"
                maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="entity" type="xs:anyURI" use="required"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Table 74: Geolocation PUSH Envelope XML schema

3.10.4.3.2 RCSPushLocation data model

Attribute	Specification	Comment
Person: <rcsenvelope> -> <rcspushlocation>	Table 74	Each client only publishes one <rcsenvelope> and one <rcspushlocation> element. The rcspushlocation element may have a label that can be used to tag the nature of the location (e.g. indicate that it's the home or provide an address, name of restaurant, etc.). If no label is provided, the location that is shared is assumed to be the sharing user's own position.
Time Zone <rcspushlocation> -> <time-offset>	Table 74, [RFC4480] and [Presence2.0_DDS]	The geolocation application may use this element to provide information on the current time zone See following chapter section for more information on the handling of the expiry of this information

Geographical Information <rcspushlocation> -> <geopriv> -> <location-info> -> <usage-rules>	Table 74, [RFC5491] and [Presence2.0_DDS]	This element can be used to provide geographical location information. The accuracy of which can be controlled by the user. See following section for more details on its encoding and on the handling of the expiry of this information
Timestamp: <rcspushlocation> -> <timestamp>	Table 74, [RFC4479]	Timestamp when the location information was pushed

Table 75: RCSPushLocation data model attributes

3.10.4.3.3 RCSLocation information

RCS clients shall not include a "*from*" attribute in the <*time-offset*> element. RCS clients shall ignore it when received.

RCS clients can provide (if authorized by the Service Provider) an "*until*" attribute in that element. The user will populate the validity time of the information with a value that will not exceed a data configuration value (see section A.1.7.2).

NOTE1: this behaviour deviates from SPI where this element is mandatory.

RCS clients shall not include the optional description attribute in the <*time-offset*> element as this overlaps with the Location Type. RCS clients shall ignore it when received.

The geographical information will be provided as geographic coordinates. As specified for the "Geographical Location" building block in [Presence2.0_DDS], encoding will use the <*geopriv*>→<*location-info*> and <*geopriv*>→<*usage-rules*> elements.

The optional <*usage-rules*> element shall contain, if present, only a "*retention-expiry*" element. The RCS client shall set the "*retention-expiry*" to the same value as the "*until*" attribute mentioned above.

NOTE2: this behaviour deviates from SPI where this element is mandatory

The <*location-info*> published by an RCS Geolocation client will contain geographical information using the GML 3.1.1 Feature Schema (see [GML3.1.1]) which is the mandatory format to be used in the <*location-info*> element. The civic location format shall not be used by RCS and location information encoded in that way will be ignored by RCS clients when received.

RCS client will within the <*location-info*> element represent an exact position by providing a GML <point> element and an inaccurate position as a <*circle*> element, both referring to the EPSG::4326 spatial reference schema as described in [RFC5491].

The coordinates of either the centre of this circle or the exact position will be represented with a single GML <*pos*> element with the actual coordinates as value.

The radius of the circle will be represented in meters, which will be indicated by setting the unit of measure attribute of the radius element to the value of EPSG::9001 as described in [RFC5491].

The text value (that is, the <*place-type*> element) shall not exceed a Service Provider configured value (see section A.1.7.2).

In case of Geolocation PUSH, the text is entered by the user.

In case of Geolocation PULL, a text can be entered automatically by the application (for example, the application, depending on the location accuracy allowed by the user, can fill a text that gives information on the user's geographical position such as street, number and

city name if a high accuracy position is allowed, or only a city name if the user only allows to provide a less precise location).

An RCS client shall ignore any other type of data provided in the *<location-info>* element.

The EPSG format requires that the coordinate representation is defined by the coordinate supplier. RCS client will always provide the coordinates in WGS 84 (latitude, longitude) decimal notion as described in [RFC5491], providing the latitude and longitude as “double”-encoded decimal numbers (as specified in [GML3.1.1]) representing the degrees, separated by a space starting with the latitude. Negative values represent Southern and Western hemisphere respectively.

The following gives an example of RCS Location information data:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<rcsenvelope xmlns=”urn:gsma:params:xml:ns:rcs:rcs:geolocation”
        xmlns:rpid=”urn:ietf:params:xml:ns:pidf:rpid”
        xmlns:gp=”urn:ietf:params:xml:ns:pidf:geopriv10”
        xmlns:gml="http://www.opengis.net/gml"
        xmlns:gs="http://www.opengis.net/pidflo/1.0"
        entity=”tel:+1234578901”>
        <rcspushlocation id=”a1233” label=”meeting location”>
              <rpid:time-offset rpid:until=”2012-03-15T21:00:00-05:00”>-300</rpid:time-offset>
              <gp:geopriv>
                    <gp:location-info>
                          <gs:Circle srsName="urn:ogc:def:crs:EPSG::4326">
                                <gml:pos>26.1181289 -80.1283921</gml:pos>
                                <gs:radius uom="urn:ogc:def:uom:EPSG::9001">10</gs:radius>
                          </gs:Circle>
                    </gp:location-info>
                    <gp:usage-rules>
                          <gp:retention-expiry>2012-03-15T21:00:00-05:00</gp:retention-expiry>
                    </gp:usage-rules>
              </gp:geopriv>
              <timestamp>2012-03-15T16:09:44-05:00</timestamp>
        </rcspushlocation>
</rcsenvelope>
```

Table 76: Example of location information data

3.10.4.4 Obtaining Location Information

A client using cellular access shall rely on the SUPL enabled terminal (SET) initiated collaboration that is specified in [SUPL] or other locating methods available from the device or network based solutions for obtaining its position. A-GPS shall be used if it has the appropriate receiver and sufficient coverage (that is, GPS satellites are visible). If it does not have this kind of receiver or if GPS positioning is not possible, a client using cellular access shall rely solely on network based positioning for obtaining its position information. In this case the positioning calculation mode is radio technology dependent, for example, for GSM (Global System for Mobile Communications)/W-CDMA (Wideband Code Division Multiple Access) networks the Location ID mode shall be used. The clients shall use the proxy mode defined in [SUPL] relying on the alternative client authentication mechanism for authentication. Support for network initiated SUPL collaboration, non-proxy mode or other authentication mechanisms described in [SUPL] is in RCS out of scope for both clients and networks, as it is not required to support the RCS use cases. The same is therefore also valid for the functions supporting this functionality (for example, the SUPL Initiation Function).

BA clients using non-cellular access can obtain location information through a regular GPS receiver if they have one available

3.10.5 NNI and IOT considerations

The NNI interfaces for geolocation services shall behave according to the procedures described in section 2.12 and the documents it refers to.

3.10.6 Implementation guidelines and examples

3.10.6.1 Geolocation PUSH

The Geolocation PUSH feature can be selected by an RCS user whenever it makes sense to share her/his location information with other RCS users, i.e.:

- From the general “share menu” or
- Inside a call / video call
- Or inside a Chat or a Group Chat

At the receiver side, a “pop up” menu advertises the user that an RCS user is communicating some location information

NOTE: for locations carrying a label, the client may offer the user to permanently store this location on the device.

3.10.6.1.1 Show in a Map

When during a call or 1-to-1 chat, a Geolocation Push is received from the conversation partner providing their location (i.e. not carrying a label as described in section 3.10.4.3.2), the client should request the user whether they want to share their location unless it was shared recently with that contact (i.e. the validity time should not have expired). If the user accepts or a valid location was already provided, the client should show both locations on a map and offer the user the option to refresh his location.

NOTE: also other locations (i.e. carrying a label) shared during the conversation may be shown on the map.

When a contact sends a location carrying the same label or carrying no label and previously a different location with the same label (or no label) was shared by that same contact, the previous location should be removed from the map and the new location should be shown instead.

Similarly if in a Group Chat the location from another participant is received, the user should be requested whether to share his location with the other participants unless that was done recently already. All received locations of other participants should be shown on a map including their own location if it was shared, with the possibility to share a refresh or initial version of the user’s own location with the other participants.

This leads to following UX:

- For the initial sender

Figure 138: Show on Map: Initiator

- For the initial recipient(s) that have not shared their location yet

Figure 139: Show on Map: initial recipients

- For a group chat participant once the user has shared their location

Figure 140: Show on Map: Group Chat participant after sharing own location

3.10.6.2 Geolocation PULL

The Geolocation PULL feature can be selected by an RCS user in same circumstance as the Geolocation PUSH feature, i.e.:

- From the general “share menu” or
- Inside a call / video call
- Or inside a chat or a Group Chat

At the receiver side:

- For the technical solution based on Geolocation API Gateway and LBS infrastructure:
  - no specific behaviour is required for the client implementation, only standard SMS is used:
    - The user is triggered by a standard SMS requesting in clear text their authorization to share their localization with the user identified by the caller number.
    - The user gives their authorization by answering their decision in a clear SMS text to a dedicated E.164 number that was communicated in the SMS that was received
    - At any time, the user can revoke the authorization using a standard SMS in clear text
- For the technical solution based on file transfer
  - A pop up menu is presented by the application when a Geolocation PULL request is received and no automatic authorization is granted to the requester (either because this is the first request received from this user or because the authorization validity time of a previously authorized request has expired.)

- The user then has the possibility to accept or deny sharing their location with the requester
- If the user accepts, they have:
  - Depending on Service Provider policy (see GEOLOCATION VALIDITY parameter defined in section A.1.7) the option to associate a validity time for the information.
  - The possibility to define the duration the authorization is granted to the Requester
  - The option to choose the level of accuracy for the location that will be provided to this requester (for example, Country, City, Street)
- At any time, in the address book, the user can activate a menu to revoke their authorization for Geolocation PULL by a dedicated contact.

3.11 Audio Messaging

3.11.1 Feature description

This feature enables an RCS user to record and/or send an audio message to his RCS contacts. An RCS Recorded Audio Message (RRAM) can be sent to one or more contacts. When the RRAM is recognized by the receiving RCS Client as being an audio message, it is handled consequently (as described further in this feature's section).

3.11.2 Interaction with other RCS features

The Audio Message feature is linked to the File Transfer service that conveys the RRAM to the recipient.

3.11.3 High Level Requirements

3-11-1 A RRAM can be sent to one or more contacts.

3-11-2 The message display will show the time, date and duration of each message.

3-11-3 Message recording shall be limited to a maximum duration

3.11.4 Technical Realization

3.11.4.1 RCS Recorded Audio Message format

An RCS client shall encode the audio message using the Adaptive Multi-Rate (AMR) codec.

The RRAM shall be formatted in the file format defined in [RFC4867].

The transport of RRAM uses the File Transfer features:

- standard notification mechanism
- store and forward when available
- auto-acceptance rules for File Transfer
- technology is either HTTP or MSRP based on the FT DEFAULT MECH configuration parameter (see Table 86) and the supported File Transfer technologies according to the capability exchange.

3.11.4.2 Sender procedures

3.11.4.2.1 Recording

When the Audio Message is selected via the User Interface, the Client shall record an audio file via the device's microphone.

The duration of the RRAM shall be limited to a maximum duration (MAX RRAM DURATION parameter as defined in section A.1.17). The Client shall automatically stop the recording when this limit is reached.

Once recorded, the content should automatically be packaged into the file format described in section 3.11.4.1.

3.11.4.2.2 Sending

When sending a RRAM to a contact, the RRAM is transported via the available File Transfer service (see section 3.5) taking into account the supported technology (i.e. MSRP or HTTP). The File Disposition shall be set to '*render*'.

NOTE '*render*' means that the content of the file can be played directly from the Chat application upon user action.

When using transport over HTTP, in complement to the procedures of section 3.5.4.8.3.1, the Client shall put the length of the RRAM in the playing-length element of the File transfer via HTTP message body content, as defined in Table 77.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:gsma:params:xml:ns:rcs:rcs:rram"
                xmlns:xs="http://www.w3.org/2001/XMLSchema"
                xmlns="urn:gsma:params:xml:ns:rcs:rcs:rram"
                elementFormDefault="qualified"
                attributeFormDefault="unqualified">
        <xs:element name="playing-length">
                <xs:simpleType>
                        <xs:restriction base="xs:integer"/>
                </xs:simpleType>
        </xs:element>
</xs:schema>
```

Table 77: Extension to File Transfer via HTTP message body schema for Audio Message

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<file xmlns="urn:gsma:params:xml:ns:rcs:rcs:fthttp"
    xmlns:am=" urn:gsma:params:xml:ns:rcs:rcs:rram“>
        <file-info type="file" file-disposition="[file-disposition]">
                <file-size>[file size in bytes]</file-size>
                <file-name>[original file name]</file-name>
                <content-type>[MIME-type for file]</content-type>
                <am:playing-length>[duration of the rram]</am:playing-length>
                <data url ="[HTTP URL for the file]" until="[validity of the file]"/>
        </file-info>
<file>
```

Table 78: Example of Audio Message Transfer using File Transfer via HTTP

3.11.4.3 Receiver procedures

On the receiving side, when a File Transfer request is received with the file-disposition set to "render" and the content is recognized as corresponding to the file format described in section 3.11.4.1, rather than announcing the transfer as a File Transfer, the UI shall announced that an audio message is received. If accepted or auto-accepted, the received content shall be displayed in the corresponding 1-to-1 or Group Chat thread as an audio message with the option to play it. The RRAM shall not be played automatically. The Display Notification (if requested) shall be sent when the playing of the file is started.

3.11.5 NNI and IOT considerations

No specific guidelines apply other than what is already defined in Section 2.12

3.11.6 Implementation guidelines and examples

From the UX point of view, two possible entry points to the Audio Message service are:

1. Address book/Call-log: An audio message can be initiated with any contact.
2. Chat window: From the Chat (one-to-one Chat only) window an audio message can be initiated using the relevant menu item. The experience is identical to the address book/call-log.

Audio messages can be shared easily by all RCS users within 1-to-1 and group chat sessions by simply holding down a soft key/button to record the message. This shall also be possible via an entry point on the contact card.

Audio messages are received within the chat or group chat thread associated with the contact that has sent the message.

3.12 Extension

3.12.1 Feature description

This feature enables an Extension to use the RCS infrastructure to communicate with other RCS entities.

3.12.2 Interaction with other RCS features

Due to its nature, the Extension feature interacts with any other RCS feature; e.g. invoke a feature.

3.12.3 High Level Requirements

An Extension shall be uniquely identified.

3-12-1 An Extension may use RCS features (e.g. File Transfer) when communicating with other RCS entities.

3-12-2 Some Extensions may require to use RCS features (e.g. Chat) only between the same instances of those Extensions.

3-12-3 An Extension may generate its own specific traffic.

3-12-4 The Extension specific traffic may be message based.

3-12-5 The Extension specific traffic may be real time based.

3-12-6 Any traffic generated by an Extension shall be identified in the network as being issued from this Extension.

3-12-7 A Service Provider shall be able to revoke an Extension.

3.12.4 Technical Realization

3.12.4.1 Communication from Extension not specifically targeted towards another specific Extension

An Extension is allowed to use any session based RCS feature just like any RCS client entity (e.g. VideoShare). In this case, when initiating a session, the standard procedures defined in the section of this document corresponding to the feature to use apply with the following modification:

When initiating a session, the SIP INVITE request initiated by the Extension shall include the Extension's IARI tag in the Contact header.

3.12.4.2 Communication between specific Extensions

This kind of communication is established only between instances of the same Extensions (i.e. they have the same IARI tag) and is only possible when the ALLOW RCS EXTENSIONS parameter (as defined in section A.1.16) is set to 1 on the Client.

As a general rule for the following sub-sections, when setting a session, the SIP INVITE requested by the Extension shall include:

- the ICSI of the service (when such ICSI is defined) in the Accept-Contact header,
- the Extension's IARI tag with the *require* and *explicit* parameters in a dedicated Accept-Contact header, provided that the Service itself is not defined by an IARI (e.g. content sharing).

  NOTE: Services defined by IARIs cannot be part of those communications between specific Extensions.

3.12.4.2.1 Communication derived from RCS services

3.12.4.2.1.1 Messaging based channel

This communication type uses the messaging ICSIs and is processed by the Messaging Server. It makes use of the standard Messaging Server processing (e.g. store and forward) with the following differences.

If the incoming SIP request contains the *require* and *explicit* parameters on the Accept-Contact header containing an IARI tag:

- The push of stored data shall be done only towards a Client hosting the same IARI as the one that has generated the stored data.
  - When storing data due to temporary unavailability of the intended recipient, the Messaging Server shall store the associated IARI tag pertaining to the IARI producing the data, including the require and explicit parameters.
  - Before pushing stored data to a reconnecting RCS Client, the Messaging Server shall check if the associated stored IARI tag is used by the reconnecting Client (e.g. via information provided by the third party register).
- No automatic storage is done in the Common Message Store.
- There is no interworking with SMS/MMS.

  NOTE: the above functionality is required to be brought into the relevant OMA specifications.

3.12.4.2.1.2 Real time based channel

No specific processing is required.

3.12.4.2.2 RCS Extension to Extension Service

3.12.4.2.2.1 Service definition

An RCS user's instance of an Extension can communicate with another RCS user's instance of an Extension (Extension to Extension) using their specific data. This data specific to the Extension is not covered in this specification. However, this specification defines a new service aiming at transporting this kind of data: the Extension to Extension service.

The Extension to Extension ICSI is defined as shown in Table 79:

Extension to Extension ICSI	Tag
Value carried in an Accept-Contact or Contact header	+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.gsma.rcs.extension"
Value carried in a P-Preferred-Service or P-Asserted-Service header	urn:urn-7:3gpp-service.ims.icsi.gsma.rcs.extension

Table 79 : Extension to Extension ICSI values

3.12.4.2.2.2 Initiating an RCS Extension to Extension Session

Handling at Initiating Nodes

The RCS Client SHALL send an initial SIP INVITE request according to the rules and procedures of [3GPP TS 24.229]. In this SIP INVITE request, the RCS Client:

1. shall include the address of the target RCS contact in the Request-URI;
2. shall include an Accept-Contact header field with the Extension to Extension ICSI 'urn%3Aurn-7%3A3gpp-service.ims.icsi.gsma.rcs.extension', percent encoded as per [3GPP TS 24.229] section 7.2A.8.2 "*Coding of the ICSI*" in a g.3gpp.icsi-ref media feature tag with the *require* and *explicit* parameters;

   NOTE: this step covers the first bullet of section 3.12.4.2.

3. shall include an Accept-Contact header field with the Extension's IARI tag with the *require* and *explicit* parameters;

   NOTE: this step covers the second bullet of section 3.12.4.2.

4. shall set the P-Preferred-Service header field with the value of the Extension to Extension ICSI 'urn%3Aurn-7%3A3gpp-service.ims.icsi.gsma.rcs.extension';
5. shall include a Contact header field with the Extension to Extension ICSI '*urn%3Aurn-7%3A3gpp-service.ims.icsi.gsma.rcs.extension*' percent encoded as per [3GPP TS 24.229] section 7.2A.8.2 "*Coding of the ICSI*" in a *+g.3gpp.icsi-ref* media feature tag, and with the Extension's IARI tag;
6. shall include the address of the originating RCS Client that has been authenticated as per section 2.5 and [3GPP TS 24.229];
7. shall include a User-Agent header field as specified in Appendix C.5 "Extension to Extension ICSI Release Version in User-Agent and Server headers";
8. should include a Session-Expires header field with the refresher parameter set to "uac" according to the rules and procedures of [RFC4028];

9. shall include a MIME SDP body as an SDP offer as described in section 3.12.4.2.2.4.
10. shall send the SIP INVITE request according to the rules and procedures of [3GPP TS 24.229]

On receipt of the SIP 200 "OK" response to the initial SIP INVITE request the RCS Client shall handle the response according to the rules and procedures of [3GPP TS 24.229], with the following clarifications:
1. The RCS Client shall start a SIP session timer using the value received in the Session-Expires header field according to the rules and procedures of [RFC4028].
2. The RCS Client shall generate and send a SIP ACK request as an acknowledgement of the final response according to the rules and procedures of [RFC3261].
3. The RCS Client shall establish the Media Plane as per [3GPP TS 24.229].

On receipt of a SIP error response to the initial SIP INVITE request the RCS Client SHALL handle the response according to the rules and procedures of [3GPP TS 24.229], with the following clarifications:
1. The RCS Client may indicate to the user that the session could not be established;
2. The RCS Client shall generate and send a SIP ACK request as an acknowledgement of the final response according to the rules and procedures of [RFC3261].

Handling at Intermediate Nodes

Intermediate nodes (e.g. access gateways, application servers) may stay in the media path depending on Service Provider policy.

Handling at Terminating Nodes

On receipt of the SIP INVITE request the RCS Client shall check if the Extension as indicated by the IARI in the Accept-Contact header is running on the device:
1. If not, the RCS Client SHALL respond with a SIP 403 Forbidden error with a Warning header set to "*Unsupported Extension*".
2. If yes, the RCS Client

    a) shall respond with a SIP 200 OK, with a valid SDP offer as per section 3.12.4.2.2.4 if the session is accepted and start a SIP session timer and take on the role of "uas" according to the rules and procedure of [RFC4028], and establish the Media Plane as per [3GPP TS 24.229], or
    b) shall respond with a SIP 603 Decline if the session is not accepted.

If the Client is already involved in an Extension to Extension session with the same contact and the same Extension (i.e. same IARI), it shall terminate the ongoing session as per section 3.12.4.2.2.3 before accepting the new one.

In a multi-device scenario, if more than one RCS Client receives the incoming SIP INVITE request because of forking by the IMS core, as per [RFC3261], only the RCS Client that responds first will remain in the session; the other session will be torn down by the IMS core. If a client responds with SIP 603 Decline, no session at all is set up as per [RFC3261]. If no RCS Client responds, the request will be timed-out.

3.12.4.2.2.3 Terminating an RCS Extension to Extension Session

To close an Extension to Extension session due to an explicit closing request from the Extension, the Client:
1. shall generate a SIP BYE request according to the rules and procedures of [3GPP TS 24.229], with the Reason Header field as defined in [RFC3326] with the protocol-value set to SIP, the protocol-cause set to *200* (e.g., *SIP;cause=200*);

2. shall send the SIP BYE request according to the rules and procedures of [3GPP TS 24.229];
3. shall release all Media Plane resources corresponding to the Extension to Extension session being closed.

A client shall close an Extension to Extension session when it has been idle for longer than the value configured for the IM SESSION TIMER configuration parameter defined in section A.1.3.3. In this case, the Client:

1. shall generate a SIP BYE request according to the rules and procedures of [3GPP TS 24.229], with the Reason Header field as defined in [RFC3326] has a protocol-value set to SIP and a protocol-cause set to *200*;
2. shall send the SIP BYE request according to the rules and procedures of [3GPP TS 24.229];
3. shall release all Media Plane resources corresponding to the Extension to Extension session being closed when a final response to that BYE request is received.

When receiving a SIP BYE request, the client shall

1. shall generate a SIP 200 OK response according to the rules and procedures of [3GPP TS 24.229];
2. shall send the SIP 200 OK response according to the rules and procedures of [3GPP TS 24.229];
3. shall release all Media Plane resources corresponding to the Extension to Extension session being closed.

NOTE: When the Extension wants to send further traffic to the other client after the session has been closed, a new session shall be started as described in section 3.12.4.2.2.2.

3.12.4.2.2.4 SDP Contents

SDP Contents when Initiating a Session

An initiating entity (e.g. an RCS Client) SHALL populate the SDP of an Extension to Extension session invitation request to match the Media Streams that are requested by the pertaining Extension. Therefore the initiating entity shall include in the SIP INVITE request a MIME SDP body as an SDP offer according to the rules and procedures of [3GPP TS 24.229]. The SDP offer shall contain media descriptions matching the requested Media Streams according to the following clarifications:

- When including an offer for a Media Stream using MSRP, the initiating entity shall include a media description according to the rules and procedures of [RFC4975] with the *a=max-size* parameter set to the RCS configuration parameter EXTENSIONS MAX MSRP SIZE (see Annex A.1.16), and *a=accept-types* shall only include the *text/plain* MIME type. Also MSRP Failure Reports shall be requested and MSRP Success Reports SHALL NOT be requested.
- When including an offer for a Media Stream for real-time continuous Media, using RTP/RTCP, the initiating entity shall include a media description according to [RFC3550], [3GPP TS 24.229], [3GPP TS 26.114] and [3GPP TS 26.141], and make use of pre-conditions.

  NOTE: for RTP there may be a need to control QoS. How this is done from the network is not covered here. Therefore Extensions making use of RTP can only be provided by Extensions known by the operator.

SDP Handling at Intermediate Nodes

Intermediate nodes shall include the contents of the SDP they received in the SDP they send out, in accordance with the rules and procedures of [3GPP TS 24.229] and [RFC3264]. Specific attributes in the SDP may be modified for the following reasons:

- To modify IP-address and port information to insert the intermediate entity in the media path of the session.

All modifications shall be done according to the rules and procedures of [RFC3264] and the respective Media Stream standards (i.e. [RFC4975] for MSRP-based media description and [RFC3264] and [RFC3550] for RCP/RTCP-based media descriptions).

SDP Handling at Terminating Nodes

A terminating entity (e.g. an RCS Client) shall process an incoming SDP and accept, modify or reject the Media Streams requested in the incoming SDP as defined by [3GPP TS 24.229] and [RFC3264]. The terminating entity SHALL handle the media descriptions according to the following clarifications:

- Media descriptions for a Media Stream for messages, using MSRP, shall be handled and responded to according to the rules and procedures of [RFC4975], with the a=max-size parameter set to the configuration parameter EXTENSIONS MAX MSRP SIZE (see Annex A.1.16), and *a=accept-types* shall only include the *text/plain* MIME type. Also MSRP Failure Reports SHALL be requested and MSRP Success Reports shall not be requested.
- Media descriptions for a Media Stream for real-time continuous Media, using RTP/RTCP, shall be handled and responded to according to the rules and procedures of [RFC3550], [3GPP TS 24.229], [3GPP TS 26.114] and [3GPP TS 26.141] and make use of pre-conditions.

  NOTE: for RTP there may be a need to control QoS. How this is done from the network is not covered here. Therefore Extensions making use of RTP can only be provided by Extensions known by the operator.

3.12.4.2.2.5 MSRP Session Handling

Clients in a session set up for MSRP shall respect the value of the a=max-size parameter from the MSRP SDP which is set using the configuration parameter EXTENSIONS MAX MSRP SIZE (see Annex A.1.16), to limit the size of content sent within the session. Also, only text/plain MIME types shall be transferred. If larger messages or files or other MIME types are to be transferred, the RCS File Transfer feature shall be used using procedures from section 3.12.4.2.1.1.

When no response is received to an MSRP SEND, the rules and procedures of [RFC4975] are followed with the following clarification:

- The client not receiving an MSRP SEND response should set the *cause=503* along with an optional protocol-text (e.g. *SIP;cause=503;text="Service Unavailable"*) in the SIP BYE request it generates. The client should indicate to the user that an error occurred when sending the message in the MSRP SEND.

3.12.4.3 Extension revocation

A Service Provider shall be able to control the use of an Extension by an RCS Client.

An Extension control request can be triggered by the network by sending a EUCR system request with type *urn:gsma:rcs:extension:control* as specified in section 2.10.4.

When the Client receives such a request, it shall take the following actions:

1. Reply to the request with a 200 OK response
2. For each (<IARI>,<duration>) pair contained in the Data attribute, block the Extension matching the <IARI> from accessing the RCS infrastructure for <duration> seconds.
   A duration of '0' means that the corresponding Extension shall no longer access the RCS infrastructure.
   No action is required from the Client in case a IARI is not locally matched with an Extension.

A new request received for an IARI already processed in a previous request shall override that previous request. An Extension which has been blocked (i.e. duration 0) can thus for example be unblocked (e.g. duration 10) via a new EUCR.

3.12.5 NNI and IOT considerations

No specific guidelines apply other than what is already defined in Section 2.12.

3.12.6 Implementation guidelines and examples

From the UX point of view, many possibilities are offered via Extension. These are out of scope of this specification.

Annex A: Managed objects and configuration parameters

This Annex provides the full details on the RCS data model including an overview of all configuration parameters. These parameters will be set using the mechanisms described in section 2.3.

The aim of this section is to provide a complete configuration data model for reference by both Service Providers and OEMs.

A.1. Management objects parameters overview

This section provides an overview of the configuration parameters used for RCS. These parameters can either come from an existing management object (like for instance the OMA defined objects for Presence, Messaging and so on) or may be RCS specific. In the latter case they will be formally defined in section A.2.

NOTE: this may not be the only document where parameters controlling an RCS device are defined (see e.g. [PRD-RCC.53]).

A.1.1. Presence related configuration

A.1.1.1. OMA Presence Provisioning parameters

OMA Presence Client provisioning parameters are defined in [PRESENCE2MO]. Table 80 lists the OMA Presence parameters applicable to RCS.

Configuration parameter	Description	RCS usage
CLIENT-OBJ-DATA-LIMIT	maximum size of the MIME object in SIP PUBLISH request	Optional parameter It is mandatory and becomes relevant only if DEFAULT DISCOVERY MECHANISM is set to PRESENCE or PRESENCE PROFILE is set to 1
CONTENT-SERVER-URI	HTTP URI of the content server to be used for content indirection	Not Used

SOURCE-THROTTLE-PUBLISH	minimum time interval (in seconds) between two consecutive publications	Optional parameter It is mandatory and becomes relevant only if DEFAULT DISCOVERY MECHANISM is set to PRESENCE or PRESENCE PROFILE is set to 1
MAX-NUMBER-OF-SUBSCRIPTIONS-IN-PRESENCE-LIST	Limits the number of back-end subscriptions allowed for a presence list. This parameter applies to the “rcs” list (as described in section 3.7.4.5)	Optional parameter It is mandatory and becomes relevant only if PRESENCE PROFILE is set to 1
SERVICE-URI-TEMPLATE	syntax of the service URI	Optional parameter It is mandatory and becomes relevant only if PRESENCE PROFILE is set to 1, and has then a value of “<xui>;pres-list=<id>” according to section 5.5.1 in [PRESENCEIG]
RLS-URI	SIP URI of the RLS to be used by the Watcher when subscribing to a Request-contained Presence List	Optional parameter

Table 80: RCS usage of OMA presence configuration parameters

A.1.1.2. RCS Specific Provisioning parameters

This RCS specification includes the following additional presence related configuration parameters:

Configuration parameter	Description	RCS usage
PRESENCE PROFILE	This parameter allows enabling or disabling the usage of the social information via presence. If set to 0, the usage of the social information via presence feature is disabled. If set to 1, the social information via presence feature is enabled. This parameter will consequently influence the inclusion of the tag associated with social information via presence in OPTIONS exchanges.	Mandatory parameter
AVAILABILITY AUTHORISATION	This parameter controls the use of Availability status feature by the device (“Allowed” or “Not Allowed” as described in section 3.7).	Optional parameter It is mandatory and becomes relevant only if PRESENCE PROFILE is set to 1.
FAVOURITE LINK CONTROL	This parameter controls how the user can set the favourite link information: automatic mode, full manual mode or a combination of those for the Favourite link where in the first case the user is can only set the favourite link from a list of predefined URLs.	Optional parameter It is mandatory and becomes relevant only if PRESENCE PROFILE is set to 1.
FAVOURITE LINK URLS	A list of pre-defined Favourite link URLs	Optional parameter It is mandatory and becomes relevant only if FAVOURITE LINK CONTROL is set to “Auto” or “Auto+Man”.
FAVOURITE LINK LABEL MAX LENGTH	This parameter allows the control of the maximum length of the label assigned to a favourite link (with a maximum value of 200 Characters).	Optional parameter It is mandatory and becomes relevant only if PRESENCE PROFILE is set to 1.

ICON MAX SIZE	This parameter allows the control of the maximum size of the picture provided in the status-icon (with a maximum value of 200KB).	Optional parameter It is mandatory and becomes relevant only if PRESENCE PROFILE is set to 1.
NOTE MAX SIZE	Maximum length of presence tagline at presentity side. The reason to have at presentity side a configurable attribute on the RCS client to control the maximum size of the Note is to make the end user aware of what the limit is (when typing the content of the Note/free text). Avoiding enforcement of this limit at network / watcher side would lead to truncating the note. This value should have a lower value than the one defined at watcher side in the OMA Presence Implementation guideline [PRESENCEIG].	Optional parameter It is mandatory and becomes relevant only if PRESENCE PROFILE is set to 1.
LOCATION TEXT MAX LENGTH	This parameter allows the control of the maximum length of the text describing the current location (with a maximum value of 200 characters).	Optional parameter It is mandatory and becomes relevant only if PRESENCE PROFILE is set to 1.
LOCATION VALIDITY	This parameter allows controlling the maximum time during which a location information should be considered valid.	Optional parameter It is mandatory and becomes relevant only if PRESENCE PROFILE is set to 1.
MAX LOCATION UPDATE	This parameter controls the minimum duration between consecutive location updates.	Optional parameter It is mandatory and becomes relevant only if PRESENCE PROFILE is set to 1.
PUBLISH EXPIRY TIME	This parameter allows setting the default expiry time for a SIP PUBLISH as described in section 2.6.1.2.2. If not present, the behaviour of a UE with a Presence client shall be to omit the Expires header from PUBLISH requests except when deleting/unpublishing an earlier presence document. In that case Expires:0 shall be included in the PUBLISH request.	Optional parameter

NON-VIP CONTACTS POLL MAX FREQUENCY	This parameter controls the maximum number of poll operations on the non-VIP contacts list during a certain period of time.	Optional parameter It is mandatory and becomes relevant only if PRESENCE PROFILE is set to 1.

Table 81: RCS additional presence related configuration parameters

A.1.2. XDM related configuration

A.1.2.1. OMA XDM Provisioning parameters

OMA XDM Client provisioning parameters are defined in [XDMMO]. The following table lists the OMA XDM parameters applicable to RCS. The mandatory parameters become optional if no functionality depending on XDM is deployed (that is no Presence based capability check as described in section 2.6.1.2 and no Social Presence as described in section 3.7 or PS Voice or Video calls as described in section 3.8 and 3.9), or if the device is VoLTE or VoHSPA enabled. VoLTE or VoHSPA devices would use the default XCAP Root URI value as defined in [PRD-IR.92] or [PRD-IR.58] respectively, but the default value could still be overwritten with the OMA XDM parameter.

Configuration parameter	Description	RCS usage
XCAP Root URI	The root of all XCAP (XML configuration access protocol) resources (which points to the Aggregation Proxy address). This is used when accessing via XCAP.	Mandatory parameter
XCAP Authentication user name	HTTP digest “username”, for accessing an XDMS (XDM server) using the XCAP protocol	Optional parameter It is mandatory and becomes relevant only if XCAP Authentication Type is set to “Digest”.
XCAP Authentication Secret	HTTP digest password	Optional parameter It is mandatory and becomes relevant only if XCAP Authentication Type is set to “Digest”.

XCAP Authentication type	Authentication method for XDMS over XCAP. Possible values: Early IMS (IP Multimedia Subsystem) or Digest. NOTE: The Early IMS value is a specific RCS value that is not defined in OMA. The support is provided according to [3GPP TS 33.141] Annex D and either sections 6.3 and 6.4 in [XDM1.1_Core] or section 5.1 in [XDM2.0_Core]. That means that in the HTTP GET request to the Aggregation Proxy the client shall supply the "*X-3GPP-Intended-Identity*" header to indicate the user identity, as specified in [3GPP TS 33.141] Annex D.	Mandatory parameter

Table 82: RCS usage of OMA XDM configuration parameters

A.1.2.2. RCS Specific Provisioning parameters

This RCS specification includes the following additional XDM related configuration parameters:

Configuration parameter	Description	RCS usage
REVOKE TIMER	This parameter allows setting the duration during which a contact should remain in the “rcs_revokedcontacts” list (as described in section 3.7.4.5). It may also be used for the frequency that the list is checked.	Optional parameter It is mandatory and becomes relevant only if PRESENCE PROFILE is set to 1.
PNB MANAGEMENT	This parameter allows to enable (1) or disable (0) the PNB feature (as described in section 2.15.1)	Optional parameter. If not present, it is assumed that PNB feature is disabled.
XDM CHANGES SUBSCRIPTION	This parameter allows to enable (**1**) or disable (**0**) the subscriptions to XDM document changes (as described in section 2.14.2.1)	Optional parameter. If not present, it is assumed that no subscriptions to XDM document changes are required

Table 83: RCS additional XDM related configuration parameters

A.1.3. Chat related configuration

A.1.3.1. OMA SIMPLE IM Provisioning parameters

OMA SIMPLE IM client provisioning parameters are defined in [RCS5-SIMPLEIM-ENDORS]. Following table only lists which of those SIMPLE IM application parameters are applicable.

Configuration parameter	Description	RCS usage
PRES-SRV-CAP	Flag used for the Messaging Server to indicate the Presence publish capability of a Presence information element of the Messaging Server on behalf of the SIMPLE IM Client	Not Used. Always set to the OMA value indicating that the capability is not supported in the network
MAX_AD-HOC_GROUP_SIZE	Maximum number of Participants allowed for an Ad-hoc Group Chat session	Optional parameter It is mandatory and becomes relevant only if CONF-FCTY-URI is set to a value different from sip:foo@bar.
CONF-FCTY-URI	SIP URI used for setting up an Ad-hoc Group or extending a 1-1 Chat session. Presence of a dummy URI ("sip:foo@bar") in the CONF-FCTY-URI parameter implies that the RCS client is not allowed to start a Group Chat	Optional parameter It is mandatory and becomes relevant only if Group Chat is enabled.
EXPLODER-URI	SIP URI used for sending SIP MESSAGE (e.g. Sending SIP MESSAGE to an Ad hoc Group)	Not Used, populated with "sip:foo@bar"
CONV-HIST-FUNC-URI	SIP URI for the SIMPLE IM user's conversation history storage	Not Used, populated with "sip:foo@bar"
DEFERRED-MSG-FUNC-URI / MSG-STORE-URI	SIP URI used for the SIMPLE IM User's message-store account for deferred messaging	Not Used, populated with "sip:foo@bar"

Table 84: RCS usage of OMA SIMPLE IM configuration parameters

A.1.3.2. OMA CPM Provisioning parameters

OMA CPM does not include any formal provisioning parameter definition. Therefore the parameters for CPM are defined as RCS specific in section A.1.3.3. Furthermore following SIMPLE IM Parameters (see section A.1.3.1) will be applicable also for CPM services:

- MAX_AD-HOC_GROUP_SIZE
- CONF-FCTY-URI
- EXPLODER-URI
- DEFERRED-MSG-FUNC-URI / MSG-STORE-URI

NOTE: if standalone messaging is enabled (see section A.1.3.3), this parameter can be set to a value different from sip:foo@bar in which case 1-to-Many standalone messaging can be used.

A.1.3.3. RCS Specific Provisioning parameters

This RCS specification includes the following additional Chat related configuration parameters:

Configuration parameter	Description	RCS usage
CHAT AUTH	This parameter Enables/Disables the Chat service. If set to **0** the Chat service is disabled. When set to **1** it is enabled.	Mandatory Parameter
GROUP CHAT AUTH	This parameter Enables/Disables the Group Chat service. If set to **0** the Group Chat service is disabled. When set to **1** it is enabled. If not present, the CHAT AUTH parameter is used to determine Group Chat enablement. If CHAT AUTH is disabled (0), GROUP CHAT AUTH shall be also disabled (0) if configured.	Optional parameter
STANDALONE MGS AUTH	This parameter Enables/Disables the Standalone Messaging Service. If set to **0** the service is disabled. When set to **1** it is enabled.	Mandatory Parameter
IM CAP ALWAYS ON	This parameter configures the client to support store and forward when presenting the Chat capability status for all the contacts. If set to **1**, the Chat capability for all RCS contacts will be always reported as available. Otherwise (**0**), the capability will be reported based on the algorithm presented in section 2.7.1.1. For example, this can be used by Service Providers that are implementing the store and forward functionality for chat on both the terminating side for its own subscribers, and the originating side for communication with subscribers belonging to other Service Providers do not have the store and forward feature.	Optional parameter (It is mandatory if CHAT AUTH is set to 1 and CONF-FCTY-URI is set.)
IM WARN SF	If IM CAP ALWAYS ON is set to enabled (use of store and forward), a new parameter is used called IM WARN SF for UI purposes only. If the IM WARN SF parameter is set to (**1**) then, when chatting with contacts which are offline (Store and Forward), the UI must warn the user of the circumstances (by showing a message on the screen for instance). Otherwise (**0**), there will not be any difference at UX level between chatting with an online or offline (Store and Forward) user.	Optional parameter (It is mandatory if CHAT AUTH is set to 1 and CONF-FCTY-URI is set and IM CAP ALWAYS ON is set to 1.)

GROUP CHAT FULL STORE FORWARD	This parameter controls whether the service provider for this client supports the Full Store and Forward feature for Group Chat (**1**) or not (**0**).	Optional parameter (It is mandatory if CHAT AUTH is set to 1 and CONF-FCTY-URI is set.)
GROUP CHAT INVITE ONLY FULL STORE FORWARD	This parameter controls whether the service provider for this client allows all users to be invited for a group chat (**0**) or only those that support the full store and forward feature for Group Chat (**1**)	Optional parameter
IM CAP NON RCS	This parameter configures the client to support chat with non-RCS contacts. If set to **1**, the Chat capability for all contacts will be always reported as available whether they are RCS enabled or not. Otherwise (**0**), the capability will be reported based on the setting for IM CAP ALWAYS ON and algorithm presented in section 2.7.1.1. For example, this can be used by Service Providers that are implementing the interworking of chat to SMS/MMS.	Optional parameter It is mandatory if CHAT AUTH and CONF-FCTY-URI is set is set and IM CAP ALWAYS ON is set to 1.
IM CAP NON RCS GROUP CHAT	This parameter defines whether and under which conditions the device is able to invite non RCS users in a Group Chat: (**0**): the device is not able to invite a non RCS user in any Group Chat session (default value) (**1**): the device is able to invite upon Group Chat creation or after Group Chat creation non RCS users in every Group Chat session (**2**): the device is able to invite upon Group Chat creation non RCS users only for the Group Chat sessions generated by its user	Optional parameter
GROUP CHAT BREAKOUT ALLOWED PREFIXES	A list of prefixes of phone numbers used to identify the non RCS contacts that the client is allowed to add in a Group Chat. The prefix is interpreted by the client by matching the phone numbers of the address book or entered by the user starting from the left. The length of the prefix can be one or more digits and it can start with the "+" character. The service provider should take the subscriber's HPLMN numbering scheme into account when defining the prefixes. For the case that no prefixes are provided, all contacts can be invited to a Group Chat without restriction.	Optional parameter It is mandatory if IM CAP NON RCS GROUP CHAT is set to 1 or 2

IM WARN IW	If IM CAP NON RCS is set to enabled (use of interworking), a new parameter is used called IM WARN IW for UI purpose only. If IM WARN IW parameter is set to (**1**) then, when chatting with non-RCS contacts (Interworking), the UI must warn the user of the circumstances. Otherwise (**0**), there will not be any difference at UX level between chatting with an online RCS or a non-RCS (SMS/MMS) user.	Optional parameter It is mandatory if CHAT AUTH and CONF-FCTY-URI is set is set to 1 and IM CAP NON RCS is set to 1.
IM SMS FALLBACK AUTH	This parameter controls whether the client automatically proposes to fall back to SMS if there is an error in transmitting a chat invite or message. If set to **0** this fallback is disabled. When set to **1** the user is proposed to send as SMS instead if there is an error.	Optional parameter (It is mandatory if CHAT AUTH is set to 1.)
IM SESSION AUTO ACCEPT	This parameter controls whether the client automatically accepts incoming session invitations (**1**) or whether acceptance depends on a user action (**0**) as defined through the IM SESSION START parameter. Automatic accept should only be used in a single device environment or if session forking on the AS is used.	Optional parameter (It is mandatory if CHAT AUTH is set to 1.)
IM SESSION START	This parameter defines the point in a chat when the receiver sends the 200 OK back to the sender confirming that the MSRP session can be established: **0** (RCS 5.x default): The 200 OK is sent when the receiver consumes the notification by opening the chat window. **1** (RCS Release 2-4 default): The 200 OK is sent when the receiver starts to type a message to be sent back in the chat window. **2**: The 200 OK is sent when the receiver presses the button to send a message (that is the message will be buffered in the client until the MSRP session is established). NOTE: as described in section 3.3.4, the parameter only affects the behaviour for a 1-to-1 session if no session between the parties has been established yet.	Optional parameter (It is mandatory if CHAT AUTH is set to 1.)
IM SESSION AUTO ACCEPT GROUP CHAT	This parameter controls whether the client automatically accepts incoming Group Chat session invitations (**1**) or whether acceptance depends on a user action (**0**) as defined through the IM SESSION START parameter. Automatic accept should only be used in a single device environment or if session forking on the AS is used.	Optional parameter (It is mandatory if CHAT AUTH is set to 1.)

FIRST MSG IN INVITE	This parameter controls whether an RCS client may include a CPIM body containing an initial message in the SIP INVITE request for setting up a session. When set to **0** such a message may not be included and the client should wait for the MSRP session to be established to send the message. When set to **1** the initial message in the chat shall be included in a CPIM body in the INVITE request. NOTE: a client shall be able to handle CPIM bodies in incoming SIP INVITE requests whatever value this parameter is set to.	Optional parameter (It is mandatory if CHAT AUTH is set to 1.)
IM SESSION TIMER	This parameter controls the time during which a 1-to-1 Chat session is allowed to be idle before it's closed. When set to **0**, there shall be no timeout. The recommended value is 3 (three) minutes.	Optional parameter (It is mandatory if CHAT AUTH is set to 1)
MAX CONCURRENT SESSIONS	This parameter controls the number of sessions that are allowed to be handled by a device. A device may not initiate or accept a new session when the current number of active sessions is equal to this maximum number. A client will therefore have to close an existing session before initiating or accepting a new one. When set to 0 this limit does not apply. NOTE: this device parameter applies only to the device. If a limit of active sessions across multiple devices is required for a user, then a parameter setting in the IMS network (HSS) for that user is required to be used.	Optional parameter (It is mandatory if CHAT AUTH is set to 1.)
MULTIMEDIA IN CHAT	This parameter controls whether or not non-text (e.g. including images) messages are allowed within the chat session. When set to **0** multimedia content may not be sent over the MSRP session associated with the chat session. The client shall also indicate this in the SDP negotiation at session set up. When set to **1**, such content may be sent and received over this MSRP channel. NOTE: When set to 0, non-text content can then be sent in a separate File Transfer session.	Optional parameter (It is mandatory if CHAT AUTH is set to 1.)
MAX SIZE 1-to-1 IM	This parameter controls the maximum size of the content sent within a 1-to-1 chat session.	Optional parameter (It is mandatory if CHAT AUTH is set to 1.)
MAX SIZE GROUP IM	This parameter controls the maximum size of the content sent within an ad-hoc Group Chat session.	Optional parameter (It is mandatory if CHAT AUTH is set to 1.)

MAX SIZE STANDALONE	This parameter controls the maximum size of a message sent as a CPM Standalone message.	Optional parameter (It is mandatory if STANDALONE MESSAGING TECHNOLOGY is set to 1.)
MESSAGE STORE URL	The URL used to access the Message Store Server The parameter is optional and if not configured, means that the Service Provider is not deploying a Message Store server.	Optional parameter
MESSAGE STORE USER / PASSWORD	The credentials to access the Message Store Server. It is an optional parameter even if MESSAGE STORE URL is configured. If it is not provided and MESSAGE STORE URL is configured, the credentials for SIP have to be used.	Optional parameter. It is mandatory if the MESSAGE STORE AUTH parameter is absent or set to “0”.
MESSAGE STORE AUTH	This parameter controls the authentication mechanism used to access the Message Store Server. **0**: Plain User Name password **1**: SASL based authentication If not provided, the authentication mechanism defaults to the same as if 0 were selected: Plain User Name password.	Optional parameter
MESSAGE STORE SYNC TIMER	This parameter controls the time interval between two client triggered synchronizations. Once any type of synchronization is triggered the timer is reset. When set to **0**, there shall be no automatic client request for synchronization.	Optional parameter (It is mandatory if MESSAGE STORE URL is configured)
SMS MESSAGE STORE	This parameter indicates to the client whether it shall store in the RCS dedicated user folder (RCSMessageStore) any sent or received SMS. If this parameter is set to **0**, client shall not store any sent or received SMS/MMS. If this parameter is set to **1**, client shall store every sent and received SMS that cannot be correlated with the Common Message Store, If this parameter is set to **2**, client shall store every sent and received SMS and shall not attempt to correlate with the Common Message Store.	Optional parameter (It is mandatory if MESSAGE STORE URL is configured)

MMS MESSAGE STORE	This parameter indicates to the client whether it shall store in the RCS dedicated user folder (RCSMessageStore) any sent or received MMS. If this parameter is set to **0**, client shall not store any sent or received MMS. If this parameter is set to **1**, client shall store every sent and received MMS that cannot be correlated with the Common Message Store, If this parameter is set to **2**, client shall store every sent and received MMS and shall not attempt to correlate with the Common Message Store.	Optional parameter (It is mandatory if MESSAGE STORE URL is configured)
CHAT MESSAGING TECHNOLOGY	This parameter allows selecting what technology is used for the chat service described in sections 3.3 and 3.4. If this parameter is set **0**, SIMPLE IM as specified in [RCS5-SIMPLEIM-ENDORS] will be used. This is the default value if the parameter is not provided. If this parameter is set **1**, CPM as specified in [RCS5-CPM-CONVFUNC-ENDORS].	Optional Parameter (It is mandatory if CHAT AUTH is set to 1.)
CHAT REVOKE TIMER	This parameter determines the maximum time between the client sending a Chat message and receiving its delivery notification. Once this timer expires without the client having received the delivery notification, the client shall automatically send a MessageRevoke request. For the case of a successful result, the user may be informed and the client shall fallback to SMS. When set to **0** (Default Value), sending MessageRevoke requests by the client is disabled.	Optional Parameter

Table 85: RCS additional Chat related configuration parameters

A.1.4. File Transfer related configuration

As there are no OMA defined parameters for File Transfer, this RCS specification includes only RCS specific parameters. These are described in the following table:

Configuration parameter	Description	RCS usage
PROVIDE FT	This parameter allows to enable (1) or disable (0) File Transfer.	Mandatory Parameter
FT MAX SIZE	This is a file transfer size limit in Kilobyte (KB). If a file is bigger than FT MAX SIZE, the transfer will be cancelled automatically. Please note that if it is set to **0**, this limit will not apply.	Optional parameter It is Mandatory if a PROVIDE FT is set to 1.
FT WARN SIZE	This is a file transfer size limit in KB to warn the user that a file transfer may end up in significant charges. Please note that if it is set to **0**, the user will not be warned.	Optional parameter It is Mandatory if a PROVIDE FT is set to 1.
FT THUMB	This parameter allows to enable (1) or disable (0) the File Transfer Thumbnail.	Optional parameter

FT STANDFWD ENABLED	This parameter allows enabling (1) or disabling (0) of the Store and Forward feature.	Optional parameter
FT CAP ALWAYS ON	This parameter describes whether the file transfer via MSRP can take place independently of whether or not the receiving end is registered: • (0 – *or not set*) File transfer depends on known capabilities of the recipient; • (1) RCS Messaging Server based store and forward can be assumed to be enabled	Optional parameter; Mandatory if FT STANDFWD ENABLED is set to 1.
FT AUT ACCEPT	This parameter controls whether the client automatically accepts incoming File Transfer invitations (**1**) or whether acceptance depends on the user explicitly accepting (**0**). The parameter is only used if the file to be transferred is smaller than the limit configured in FT WARN SIZE. For files that are larger, the invitation will always require manual acceptance. Automatic accept should only be used in a single device environment or if session forking on the AS is used.	Optional parameter It is Mandatory if a PROVIDE FT is set to 1.
FT HTTP CS URI	This parameter configures the URI of the HTTP content server where files will be uploaded by the originating side in case the destination cannot accept within the validity period. The parameter shall contain a full qualified URI. The URI should contain the "https" schema to enforce use of secure connections for the client's content server transactions.	Optional parameter
FT HTTP CS USER	This parameter is the name or identity that shall be used to authenticate the RCS client trying to either get a root URL (HTTP GET request) or upload a file (HTTP post request).	Optional parameter; Mandatory if FT HTTP CS URI is set
FT HTTP CS PWD	This parameter is the password that shall be used to authenticate the RCS client trying to either get a root URL (HTTP GET request) or upload a file (HTTP post request).	Optional parameter; Mandatory if FT HTTP CS USER is set
FT DEFAULT MECH	This parameter controls which file transfer mechanism (MSRP or HTTP) shall be used if both ends support both mechanisms	Optional parameter; Mandatory if FT HTTP CS URI is set.

Table 86: RCS additional File Transfer related configuration parameters

A.1.5. Content Sharing related configuration

As there are no OMA defined parameters for content sharing, this RCS specification includes only RCS specific parameters. These are described in the following table:

Configuration parameter	Description	RCS usage
PROVIDE VS	This parameter allows to enable (1) or disable (0) Video Share.	Mandatory Parameter
PROVIDE IS	This parameter allows to enable (1) or disable (0) Image Share.	Mandatory Parameter
ALLOW VS SAVE	This parameter allows a Service Provider to configure whether a Video or Image Share session initiated by the RCS client can be saved or not. When set to (-1) the inclusion of the attribute defined in section 3.6.4.1.3 is up to user preference, when set to (0) the attribute will never be included, which is also the default handling if not provided, when set to (1) the attribute will always be included. NOTE: The parameter name includes VS for historic reasons.	Optional Parameter
VS MAX DURATION	This parameter enables the Service Provider of the inviting user's RCS client to control the maximum duration time of a Video Share session that the inviting user's RCS client is authorized to handle.	Optional parameter It is Mandatory if a PROVIDE VS is set to 1.
IS MAX SIZE	Maximum authorized size of the content that can be sent within an Image Share session. This parameter enables the Service Provider of the inviting user's RCS client to control the maximum size of the content that the inviting user's RCS client is authorized to send in an Image Share session.	Optional parameter It is Mandatory if a PROVIDE IS is set to 1.

Table 87: RCS additional content sharing related configuration parameters

A.1.6. IMS Core / SIP related configuration

A.1.6.1. VoLTE/VoHSPA Enabled device configuration

In a device enabled for VoLTE/VoHSPA (see section 2.2.1), the default IMS settings as defined in [PRD-IR.92] or [PRD-IR.58] are expected to be used, so the IMS Core/SIP related configuration would not be required.

For example, the own SIP or tel URI will not be configured through the management object referred to in section A.1.6.2, but rather be received in the 200 OK response to the SIP REGISTER request and the SIP Proxy is provided in Protocol Configuration Options (PCO) information received during Packet Data Protocol (PDP) context activation.

A.1.6.2. RCS endorsement of 3GPP IMS Management Object (MO)

Basic IMS/SIP client parameters are defined in 3GPP TS "IMS 3GPP IMS Management Object (MO)" [3GPP TS 24.167]. They do not directly depend on RCS, but correct settings of these parameters are essential for RCS operation. They are populated by the Service Provider according to the deployment conditions of the IMS core network providing access to RCS services.

Also, it should be noted that:

- Both a SIP and a tel URI may be configured for a user with following clarifications:

  - The configured values should not be used in the non-REGISTER transactions; instead the client uses one of the SIP or tel URIs provided in the P-Associated-URI header field returned in the 200 OK to the SIP REGISTER request as described in [3GPP TS 24.229]
  - The user's own tel URI and/or SIP URI identities are configured through the Public_user_identity parameters defined in [3GPP TS 24.167][34].
  - The public identity used for IMS registration is built according to the procedure defined in [3GPP TS 24.229].
  - When the device has either ISIM or USIM present and the RCS client has access to the ISIM or USIM, it does not rely on the SIP URI and tel URI configuration parameters.
  - If the device has neither ISIM nor USIM present or is not able to access to it, a SIP URI must be configured. This URI is used for REGISTER transactions.
  - Configuration of the tel URI is optional
- The SIP proxy is configured through the parameters hosted by the LBO_P-CSCF_Address sub-tree defined in [3GPP TS 24.167]. When the P-CSCF address has an "FQDN" type, the procedure described in section 2.4.7 applies. When the P-CSCF address has an "IP Address" type, the SIP transport protocol should be selected based on Service Provider customized settings.

A.1.6.3. RCS Specific Provisioning parameters

This RCS specification includes the following additional IMS Core/SIP related configuration parameters:

Configuration parameter	Description	RCS usage
IMS Mode Authentication Type	Specifies the type of authentication support for SIP. NOTE: In "IETF" Digest authentication is assumed. Accepted values are: • Early IMS • IMS AKA • SIP DIGEST (without TLS)	Mandatory Parameter, NOTE: a VoLTE enabled device always uses IMS AKA when in cellular PS access and can ignore this parameter

[34] The private identity (*Private_user_identity*), public identity (*Public_user_identity_List/<X>/Public_user_identity*) and domain (*Home_network_domain_name*) objects mentioned in [3GPP TS 24.167] are defined as read-only and these parameters should be obtained by the UE using the procedures described [3GPP TS 24.229]. This specification makes and exception to that definition and considers them writable during the autoconfiguration process (OMA-DM or the alternative HTTP mechanism).

Realm	Realm to use for authentication (Digest mode only)	Optional parameter It is Mandatory if a IMS Mode Authentication Type is set to Digest.
Realm User Name	Realm username to use for authentication (Digest mode only)	Optional parameter It is Mandatory if a IMS Mode Authentication Type is set to Digest.
Realm User Password	Realm user password to use for authentication (Digest mode only)	Optional parameter It is Mandatory if a IMS Mode Authentication Type is set to Digest.
tel or SIP URI – international	Specifies whether telephone numbers in international format shall in outgoing SIP requests be sent as tel URIs [RFC3966]or as SIP URIs with “user”-parameter set to “phone” as defined in [RFC3261] See Section 2.5.3.1	Mandatory Parameter
tel or SIP URI - for non- international format	Specifies whether telephone numbers in non-international format shall in outgoing SIP requests be sent as tel URIs [RFC3966] or as SIP URIs with “user”-parameter set to “phone” as defined in [RFC3261] See Section 2.5.3.1	Mandatory Parameter
Register Q-value	Q-value in Contact parameter in SIP Register, may be used in a multi-terminal deployment to control forking of incoming SIP requests, but it is not recommended to be used because it can unintentionally affect service-based routing. Recommended value: 1.0 (Note that if the parameter is not present, no Q-value will be sent in the SIP REGISTER, and the network will use the default value of 1.0.)	Optional Parameter NOTE: this parameter was Mandatory in RCS 5.0 but is now optional.

Table 88: RCS additional IMS Core/SIP related configuration parameters

A.1.7. Geolocation related configuration

A.1.7.1. OMA SUPL Provisioning parameters

RCS uses SUPL [SUPL] for providing localization social presence information.

SUPL Client provisioning parameters are defined in OMA Management Object for SUPL [SUPLMO]. RCS may use this object for provision of the required parameters for accessing the H-SLP (Home SUPL Location Platform).

Following table lists the OMA SUPL parameters applicable to RCS. The mandatory parameters become optional if no functionality depending on SUPL is deployed (that is Social Presence as described in section 3.7 or the location functionality described in section 3.10).

Configuration parameter	Description	RCS usage
Addr	The address of the H-SLP	Mandatory parameter
AddrType	The type of the address provided in Addr	Optional parameter

Table 89: RCS usage of OMA SUPL configuration parameters

A.1.7.2. RCS Specific Provisioning parameters

This RCS specification includes the following additional geolocation related configuration parameters

Configuration parameter	Description	RCS usage
PROVIDE GEOLOC PUSH	This parameter allows enabling (**1**) or disabling (**0**) the Geolocation PUSH service.	Mandatory Parameter
PROVIDE GEOLOC PULL	This parameter allows to disable (0) the Geolocation PULL service, enable it with only LBS technology (1), enable it with only File Transfer technology (2), enable it with both technologies with priority to LBS technology (3), or enable it with both technologies with priority to File Transfer technology (4).	Mandatory Parameter
GEOLOCATION TEXT MAX LENGTH	This parameter allows the control of the maximum length of the text describing the current location (with a maximum value of 200 characters.	Optional parameter It is mandatory and becomes relevant only if the Geolocation PULL or Geolocation PUSH service is available for the device.
GEOLOCATION VALIDITY	This parameter allows controlling the maximum time during which a location information should be considered valid.	Optional parameter. If present, it indicates A maximum value the user is authorized to enter.
GEOLOCATION PULL OPEN	This parameter describes whether Geolocation PULL can be used to attempt to obtain the location of all subscribers (1, i.e. of both RCS and non-RCS users) or only of RCS users that have indicated the capability (0).	Optional parameter. If it is not provided Geolocation PULL can only be used with as target RCS users that have the capability.
GEOLOCATION PULL API GW	Provides the address of the API GW to be used to send Geolocation PULL API requests	Optional Parameter It is mandatory and becomes relevant only if the Geolocation PULL with LBS service is available for the device.

GEOLOCATION PULL BLOCK TIMER	The interval during which the Geolocation PULL application is not allowed to send a PULL request to a target contact if a previous request was explicitly rejected	Optional Parameter It is mandatory and becomes relevant only if the Geolocation PULL with File Transfer service is available for the device.

Table 90: RCS additional geolocation related configuration parameters

A.1.8. Configuration related with Address book Back-up/Restore

This RCS specification does not include any additional address book back-up/restore related configuration parameters.

A.1.9. Configuration related to secondary devices

A.1.9.1. General

With the Introduction of the broadband secondary device in RCS, there are features in a broadband RCS device that require configuration:

- Control of service delivery:
  Control of service delivery: in a broadband RCS device, as specified in section 2.11.2, this user control facility is itself controlled by the Service Provider that may define the set of services subject to this function
- SMS over IP:
  As specified in [PRD-IR.92], when sending a short message from the RCS Broadband Access (BA) client, the address of the Service Provider's SMS-C needs to be supplied in the SIP request containing the short message, see [3GPP TS 24.341] chapter 5.3.1.
- MMS:
  Before sending a multimedia message from the RCS client, and when retrieving the multimedia message, the addresses of the Service Provider's HTTP proxy and MMS-C (Multimedia Messaging Service Centre) needs to be configured.

A.1.9.2. Specific RCS Configuration parameters for Control of service delivery

Network authorization for user controlling delivery of

- Voice Calls
- Video Calls
- Chat
- Sending SMS
- File Transfer
- Video Sharing
- Image Sharing
- Geolocation PUSH

  NOTE: Geolocation PULL is provided on the primary device only and as such not subject to control of service delivery

A.1.9.3. RCS endorsement of OMA MMS parameters

MMS client provisioning parameters are defined in OMA Management Object for MMS [MMSMO]. RCS BA clients may use this object for provision of the required parameters for accessing the MMS service.

Specifically, the URL to the MMS-C (MMS Proxy-Relay server) shall be provided.

A.1.9.4. RCS endorsement of OMA Connectivity Management Objects parameters

SMS-C Address: a public service identifier (PSI) in form of a tel URI or SIP URI

The NAP (network access point) object defined in [CONNMO] may be used for this purpose.

Specifically the address type field and the address field shall be provided (with SMS-C address information).

HTTP proxy Client provisioning parameters are defined by the “proxy” object in [CONNMO] and further specified in [CONNMOHTTP]. RCS Broadband access clients may use this object for provision of the required parameters for accessing the HTTP proxy.

Specifically, the proxy type, proxy address and the authorization type and credentials (username & password) shall be provided.

A.1.10. Capability discovery related configuration

This RCS specification includes the following RCS Specific configuration parameters related to the capability discovery:

Configuration parameter	Description	RCS usage
POLLING PERIOD	This is the frequency in seconds at which to run a periodic capabilities update for all the contacts in the phone’s address book whose capabilities are not available (such as non-RCS users) or are expired (see CAPABILITY INFO EXPIRY parameter). Please note that if set to **0**, this periodic update is not/no longer performed.	Mandatory parameter
POLLING RATE	This parameter allows controlling the maximum rate at which SIP OPTIONS and Presence Fetch operations are performed for all contacts combined. It therefore provides some control over the network load caused when performing a capability discovery for the whole address book.	Optional parameter (It is mandatory if POLLING PERIOD is set to a value greater than 0.)
POLLING RATE PERIOD	This parameter allows defining the window in seconds over which the configured polling rate should be measured.	Optional parameter (It is mandatory if POLLING PERIOD is set to a value greater than 0)

CAPABILITY INFO EXPIRY	When using the capability discovery mechanism and with the aim of minimizing the traffic, an expiry time is set in the capability information fetched using SIP OPTIONS or Presence. • When performing a whole address book capability discovery (i.e. polling), a capability query takes place only if the time since the last capability update took place is greater than this expiration parameter. • Default value: 2592000 (30 days)	Optional parameter (It is mandatory if POLLING PERIOD is set to a value greater than 0.)
CAPABILITY DISCOVERY MECHANISM	This parameter allows selecting the default capability and new user discovery mechanism. If not provided or set to OPTIONS, the default mechanism employed for capability discovery and new users will be OPTIONS. Otherwise (PRESENCE), it will relay of presence-based discovery by default.	Mandatory parameter
CAPABILITY DISCOVERY VIA COMMON STACK	This parameter allows selecting whether the device will fall back to OPTIONS if a discovery using presence fails with an error indicating that the other user does not support a presence based capability check. When set to **1**, this fallback is done. When set to **0** it is not done.	Optional parameter It is mandatory if CAPABILITY DISCOVERY MECHANISM is set to PRESENCE.
CAPABILITY DISCOVERY ALLOWED PREFIXES	A list of prefixes of phone numbers used to identify the contacts that are considered for the capability discovery mechanism. In case no prefix is included, capability discovery applies to all contacts. The prefix is interpreted by the client by matching the phone numbers of the address book or entered by the user starting from the left. The length can be one or more digits and it can start with the "+" character. The service provider should take the subscriber's HPLMN numbering scheme into account when defining the prefixes.	Optional parameter
NON RCS CAPABILITY INFO EXPIRY	This parameter allows to better control the amount of capability query sent to non RCS contacts. When updating a capability for a non RCS contact, a capability query takes place only if the time since the last capability update took place is greater than this parameter. Default value: 2592000 (30 days)	Optional parameter

Table 91: RCS additional capability discovery related configuration parameters

A.1.11. APN configuration

This RCS specification includes the following RCS Specific configuration parameters targeting APN configuration (see sections 2.9.1.4 and 2.13):

Configuration parameter	Description	RCS usage
RCS-E ONLY APN	This is the reference/identifier of the APN configuration which should be used to provide PS connectivity ONLY to RCS as described in section 2.9.1.4.	Mandatory parameter
ENABLE RCS-E SWITCH	As described in section 2.9.1.4 the user shall be able configure to allow or disallow RCS and/or internet traffic in the device settings. If this parameter is set to 1, the setting is shown permanently. 0, the setting is only shown during roaming. -1: RCS Switch is never shown.	Mandatory parameter
ALWAYS USE IMS APN	This parameter controls the use of the IMS APN when the device that can support the IMS APN is not in RCS-VoLTE or RCS-VoHSPA mode: it should be used always when available (1), other non-cellular connections can be used for RCS even when the IMS APN is available (0, default value) or the IMS APN is never used (-1). NOTE1: a device that cannot support the IMS APN shall ignore this parameter NOTE2: the value of -1 shall not be used for a device that is configured to support RCS-VoLTE or RCS-VoHSPA mode when available	Optional Parameter

Table 92: RCS roaming configuration parameters

A.1.12. End User Confirmation parameters

This RCS specification includes the following RCS Specific configuration parameters targeting the End User Confirmation configuration (see section 2.10):

Configuration parameter	Description	RCS usage
END USER CONF REQ ID	This is the URI that is used to identify the sender of the End User Confirmation Requests.	Optional Parameter

Table 93: RCS end user confirmation configuration parameters

A.1.13. Multidevice configuration parameters

This RCS specification includes the following RCS Specific configuration parameters targeting the multidevice configuration when using the sip.instance approach described in section 2.4.2 and 2.11:

Configuration parameter	Description	RCS usage
uuid_Value	This is the UUID value used for sip.instance	Optional Parameter

Table 94: RCS multidevice configuration parameters

A.1.14. IP Voice and Video Call configuration

As there are no OMA defined parameters for IP Voice and Video Call, this RCS specification includes only RCS specific parameters. These are described in the following table:

Configuration parameter	Description	RCS usage
PROVIDE IR94 VIDEO	This parameter allows to enable (1) or disable (0) IR94 Video Calling	Optional Parameter. This parameter SHOULD be set if IR94 client is present on UE
PROVIDE RCS IP VOICE CALL	This parameter allows to enable or to disable the RCS IP Voice Call Service on devices in RCS-CS and RCS-AA mode depending on network connectivity (only non-3GPP/non-3GPP2 networks, also on LTE, etc.).	Mandatory Parameter
PROVIDE RCS IP VIDEO CALL	This parameter allows to enable or to disable the RCS IP Video Call Service on devices in RCS-CS and RCS-AA mode depending on network connectivity (only non-3GPP/non-3GPP2 networks, also on LTE, etc.).	Mandatory Parameter
RCS IP VOICE CALL BREAK OUT	This parameter indicates to a device in RCS-AA mode whether it can use the RCS IP Voice Call service to reach any user or only RCS users that have indicated the corresponding capability.	Optional Parameter (It becomes mandatory if the device is capable of functioning in RCS-AA mode and PROVIDE RCS IP VOICE CALL is not disabled.)
RCS IP VOICE CALL BREAK OUT CS	This parameter indicates to a device in RCS-CS mode whether it can use the RCS IP Voice Call service to reach any user or only RCS users that have indicated the corresponding capability.	Optional Parameter (It becomes mandatory if the device is capable of functioning in RCS-CS mode and PROVIDE RCS IP VOICE CALL is not disabled.)

RCS IP VIDEO CALL UPGRADE FROM CS	This parameter indicates to a device in RCS-CS mode whether it is allowed to offer the upgrade a CS call to an RCS IP Video Call service if the other party in the call indicates the corresponding capability.	Optional Parameter (It becomes mandatory if the device is capable of functioning in RCS-CS mode and PROVIDE RCS IP VIDEO CALL is not Disabled.)
RCS IP VIDEO CALL UPGRADE ATTEMPT EARLY	This parameter indicates to a device that supports RCS-CS mode whether it can initiate an RCS IP Video Call upgrade without first tearing down the CS voice call.	Optional Parameter (It becomes mandatory if the device supports RCS-CS mode and PROVIDE RCS IP VIDEO CALL is not disabled.)
RCS IP VIDEO CALL UPGRADE ALLOWED ON CAPABILITY ERROR	This parameter indicates to a device supporting RCS-AA or RCS-CS mode whether it can initiate an RCS IP Video Call upgrade even if service capability exchange fails with 480 Temporarily Unavailable or 408 Timeout.	Optional Parameter (It becomes mandatory if the device supports RCS-CS mode and PROVIDE RCS IP VIDEO CALL is not disabled.)

Table 95: RCS IP Voice and Video Call configuration parameters

A.1.15. Service Provider specific extensions

A Service Provider may provide Service Provider specific extensions to the configuration parameters. This can be done both at the individual service level and add the global level (e.g. for the configuration of Service Provider specific services). All parameters are optional and if provided may be ignored by clients that are not Service Provider specific.

A.1.16. Extensions configuration parameters

The RCS specification includes the following additional Extensions related configuration parameters:

Configuration parameter	Description	RCS usage
ALLOW RCS EXTENSIONS	This parameter indicates to a device whether Extensions using the RCS infrastructure are allowed or not. If this parameter is set to: 0, Not Allowed: The SIP REGISTER shall NOT include the IARIs pertaining to Extensions, and shall NOT include the ICSI for RCS Extension to Extension in its Contact header. Also, they are not included as part of capability discovery. 1, Allowed: The SIP REGISTER shall include IARIs pertaining to Extensions, and the ICSI for RCS Extension to Extension if used by installed applications, in its Contact header. Also, they are included as part of capability discovery.	Mandatory Parameter
EXTENSIONS MAX MSRP SIZE	This parameter controls the maximum size of the MSRP content sent within a 1-to-1 RCS Extension to Extension session. A value of **0** (zero) means the maximum size of content sent is unlimited.	Optional parameter (It is mandatory and becomes relevant only if ALLOW RCS EXTENSIONS is set to **1**

Table 96: RCS extensions configuration parameters

NOTE: parameters controlling the use of the terminal APIs on an RCS device may be defined in [PRD-RCC.53]

A.1.17. Audio Messaging configuration parameters

The RCS specification includes the following additional Audio Messaging related configuration parameters

Configuration parameter	Description	RCS usage
MAX RRAM DURATION	This parameter indicates to a device the maximum duration of an RCS Audio Message in seconds. Default Value is 10 minutes. Value of 0 means no limitation.	Optional Parameter

Table 97: RCS Audio Messaging configuration parameters

NOTE: The resulting file size should not exceed the maximum file size as defined in FT MAX SIZE

A.2. RCS Management trees additions

Please note that all the configuration subtrees described in this section have as type property for the root nodes (that is the /<X> root nodes) urn:gsma:mo:rcs:5.2. All RCS specific MOs shall be placed in this RCS subtree:

Node: /<x>

Under this interior node the RCS parameters that belong to RCS specific MOs are placed

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 98: RCS MO sub tree addition node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs:5.2*
- Associated HTTP XML characteristic type: N/A

The DM Client assigns a unique name for the <x> node which consists of *[Parental Node]-Index* (e.g. *Ext/<x>/ChatAuth -> Ext/Ext-1/ChatAuth*). The index 1 belongs to the actual SIM card.

The following alert type MUST be used in a Generic Alert [DMPRO] message sent by the DM client in case of a client initiated management session towards the DM server related to an RCS MO:

- *urn:gsma:mo:rcs:5.2:provision*

The alert type is used to identify the operation that needs to be performed on the device and identifies the current version of the RCS Management Object.

A.2.1. Services sub tree additions

This RCS specification includes the following additions as a new services sub tree, the Services MO sub tree. Please note this sub tree is not included in any other specifications. So no other nodes from those specifications need to be added:

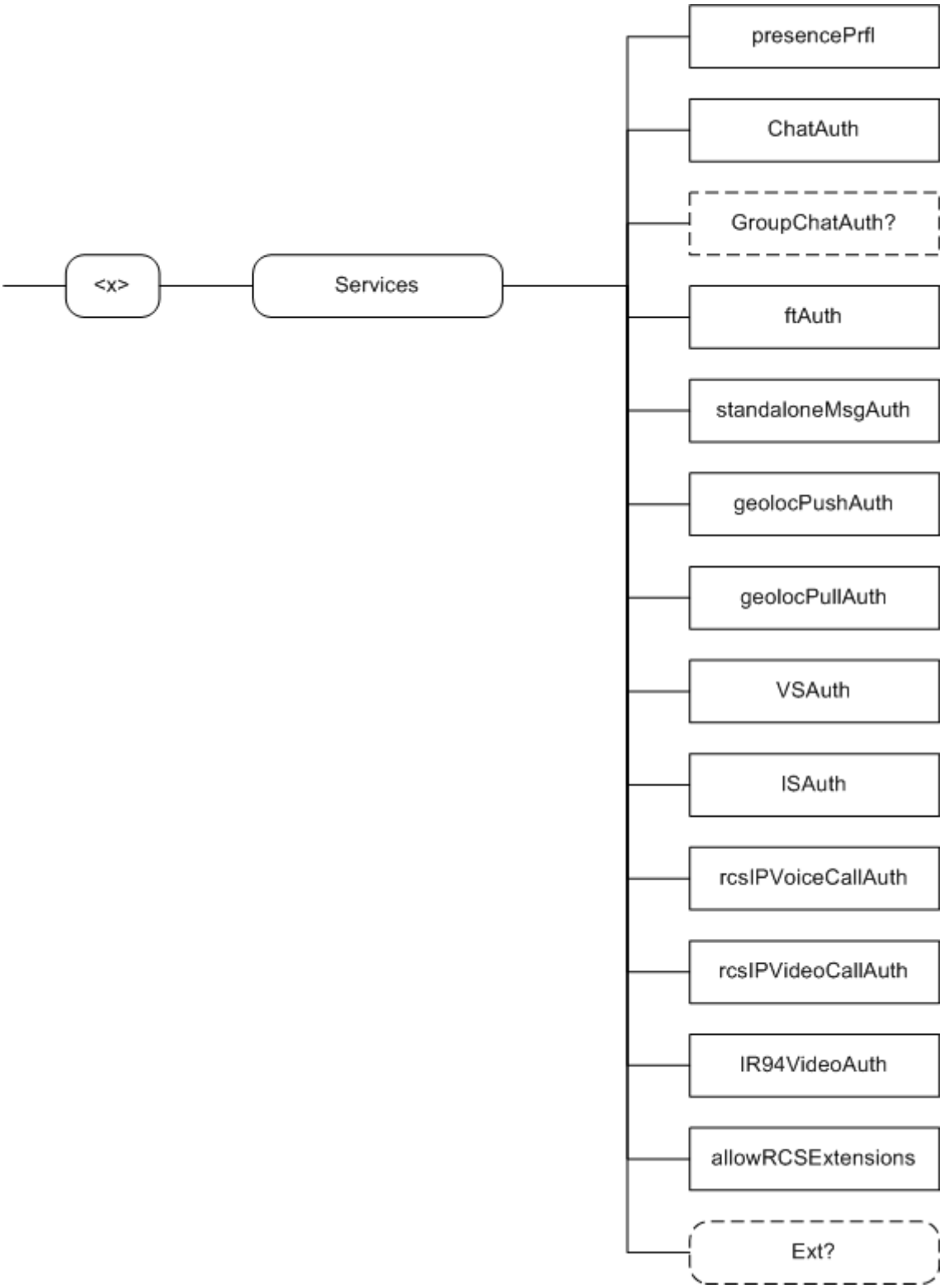


Figure 141: RCS additions, Services sub tree

The associated HTTP configuration XML structure is presented in the table below:

```
<characteristic type="SERVICES">
        <parm name="presencePrfl" value="X"/>
        <parm name="ChatAuth" value="X"/>
        <parm name="GroupChatAuth" value="X"/>
        <parm name="ftAuth" value="X"/>
        <parm name="standaloneMsgAuth" value="X"/>
        <parm name="geolocPullAuth" value="X"/>
        <parm name="geolocPushAuth" value="X"/>
        <parm name="vsAuth" value="X"/>
        <parm name="isAuth" value="X"/>
        <parm name="rcsIPVoiceCallAuth" value="X"/>
        <parm name="rcsIPVideoCallAuth" value="X"/>
        <parm.name="IR94VideoAuth" value="X"/>
        <parm.name="allowRCSExtensions" value="X"/>
        <characteristic type="Ext"/>
</characteristic>
```

Table 99 : Services MO sub tree associated HTTP configuration XML structure

Node: /<x>/Services

Under this interior node the RCS parameters related to the enabling/disabling of services are placed

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 100: Services MO sub tree addition services node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-services:5.2*
- Associated HTTP XML characteristic type: "SERVICES"

Node: /<x>/Services/presencePrfl

Leaf node that describes whether or not the social presence functionality is supported

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get, Replace

Table 101: Services MO sub tree addition parameters (presencePrfl)

- Values: If set to 1, it is supported. If set to 0, it is not supported.
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter: "presencePrfl"

Node: /<x>/Services/ChatAuth

Leaf node that represents the authorization for the user to use the chat service

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get, Replace

Table 102: Services MO sub tree addition parameters (ChatAuth)

- Values: 0, 1
  0- Indicates that chat service is disabled
  1- Indicates that chat service is enabled

- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter ID: “ChatAuth”

Node: /<x>/Services/GroupChatAuth

Leaf node that represents the authorization for the user to use the group chat service

If not instantiated, the ChatAuth parameter shall control the authorization for both 1-to-1 and Group Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get, Replace

Table 103: Services MO sub tree addition parameters (GroupChatAuth)

- Values: 0, 1
  0- Indicates that Group Chat service is disabled
  1- Indicates that Group Chat service is enabled
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter ID: “GroupChatAuth”

Node: /<x>/Services/ftAuth

Leaf node that represent the authorization for user to use the File Transfer service

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get, Replace

Table 104: Services MO sub tree addition parameters (ftAuth)

- Values: 0, 1
  0- Indicates that File Transfer service is disabled
  1- Indicates that File Transfer service is enabled
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter ID: “ftAuth”

Node: /<x>/Services/standaloneMsgAuth

Leaf node that represents the authorization for user to use the standalone messaging service

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get, Replace

Table 105: Services MO sub tree addition parameters (standaloneMsgAuth)

- Values: 0, 1
  0- The standalone messaging service is not provided. SMS and MMS is used instead
  1- The standalone messaging service is provided and uses CPM as specified in [RCS5-CPM-CONVFUNC-ENDORS].
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.

- Associated HTTP XML parameter ID: “standaloneMsgAuth”

Node: /<x>/Services/geolocPullAuth

Leaf node that represents the authorization for the user to use the Geolocation PULL service

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get, Replace

Table 106: Services MO sub tree addition parameters (geolocPullAuth)

- Values: 0, 1, 2, 3, 4
  0- Indicates that Geolocation PULL service is disabled
  1- Indicates that Geolocation PULL service is enabled only with LBS technology
  2- Indicates that Geolocation PULL service is enabled only with File Transfer technology
  3- Indicates that Geolocation PULL service is enabled with both technologies and priority for LBS technology
  4- Indicates that Geolocation PULL service is enabled with both technologies and priority for File Transfer technology
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter ID: “geolocPullAuth”

Node: /<x>/Services/geolocPushAuth

Leaf node that represents the authorization for the user to use the Geolocation PUSH service

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get, Replace

Table 107: Services MO sub tree addition parameters (geolocPushAuth)

- Values: 0, 1
  0- Indicates that Geolocation PUSH service is disabled
  1- Indicates that Geolocation PUSH service is enabled
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter ID: “geolocPushAuth”

Node: /<x>/Services/VSAuth

Leaf node that represents the authorization for user to use Video Share service

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get, Replace

Table 108: Services MO sub tree addition parameters (VSAuth)

- Values: 0, 1
  0- Indicates that Video Share service is disabled
  1- Indicates that Video Share service is enabled
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.

- Associated HTTP XML parameter ID: “vsAuth”

Node: /<x>/Services/ISAuth

Leaf node that represents the authorization for user to use Image Share service

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get, Replace

Table 109: Services MO sub tree addition parameters (ISAuth)

- Values: 0, 1
  0- Indicates that Image Share service is disabled
  1- Indicates that Image Share service is enabled
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter ID: “isAuth”

Node: /<x>/Services/rcsIPVoiceCallAuth

Leaf node that represents the authorization for user to use RCS IP Voice Call service

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get, Replace

Table 110: Services MO sub tree addition parameters (rcsIPVoiceCallAuth)

- Values: an unsigned 32 bit integer value that is mapped to a bit array indicating the radio technologies in which an RCS IP Voice Call can be initiated. The mapping is as follows from MSB to LSB:

31 MSB	…	4	3	2	1	0 LSB
Reserved	Reserved	Reserved	LTE	HSPA	3G	Wi-Fi

Table 111: rcsIPVoiceCallAuth value to radio technology mapping

Reserved bits should be ignored by the client.
NOTE: For established calls, the call should be continued as long as there is IP continuity and the available bandwidth allows.

Some examples of this mapping of values to radio technologies in which RCS IP Voice Calls are supported (only least significant byte mentioned):

xxxx0000b (i.e. 0)- Indicates that the RCS IP Voice Call service is disabled
xxxx0001b (i.e. 1)- Indicates that the RCS IP Voice Call service is enabled for non-3GPP/non-3GPP2 access only (e.g. Wi-Fi, xDSL)
xxxx1000b (i.e. 8)- Indicates that the RCS IP Voice Call service is enabled for LTE access only
xxxx1001b (i.e. 9)- Indicates that the RCS IP Voice Call service is enabled for non-3GPP/non-3GPP2 access (e.g. Wi-Fi, xDSL) and for LTE access
xxxx1100b (i.e. 12)- Indicates that the RCS IP Voice Call service is enabled for LTE/HSPA access only
xxxx1101b (i.e. 13)- Indicates that the RCS IP Voice Call service is enabled for non-3GPP/non-3GPP2 access (e.g. Wi-Fi, xDSL) and for LTE/HSPA access
xxxx1110b (i.e. 14)- Indicates that the RCS IP Voice Call service is enabled for 3G, HSPA and LTE
00001111b (i.e. 15)- Indicates that the RCS IP Voice Call service is enabled for WiFi, 3G, HSPA and LTE cellular access

- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter ID: “rcsIPVoiceCallAuth”

Node: /<x>/Services/rcsIPVideoCallAuth

Leaf node that represents the authorization for user to use the RCS IP Video Call service

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get, Replace

Table 112: Services MO sub tree addition parameters (rcsIPVideoCallAuth)

- Values: an unsigned 32 bit integer value that is mapped to a bit array indicating the radio technologies in which an RCS IP Video Call can be initiated. The mapping is as follows from MSB to LSB:

31 MSB	…	4	3	2	1	0 LSB
Reserved	Reserved	Reserved	LTE	HSPA	3G	Wi-Fi

Table 113: rcsIPVideoCallAuth value to radio technology mapping

Reserved bits should be ignored by the client.
NOTE: For established calls, the call should be continued as long as there is IP continuity and the available bandwidth allows.

Some examples of this mapping of values to radio technologies in which RCS IP Video Calls are supported (only least significant byte mentioned):
xxxx0000b (i.e. 0)- Indicates that the RCS IP Video Call service is disabled
xxxx0001b (i.e. 1)- Indicates that the RCS IP Video Call service is enabled for non-3GPP/non-3GPP2 access only (e.g. Wi-Fi, xDSL)
xxxx1000b (i.e. 8)- Indicates that the IP Video Call service is enabled for LTE access only
xxxx1001b (i.e. 9)- Indicates that the RCS IP Video Call service is enabled for non-3GPP/non-3GPP2 access (e.g. Wi-Fi, xDSL) and for LTE access
xxxx1100b (i.e. 12)- Indicates that the RCS IP Video Call service is enabled for LTE/HSPA access only
xxxx1101b (i.e. 13)- Indicates that the RCS IP Video Call service is enabled for non-3GPP/non-3GPP2 access (e.g. Wi-Fi, xDSL) and for LTE/HSPA access
xxxx1110b (i.e. 14)- Indicates that the RCS IP Video Call service is enabled for 3G, HSPA and LTE
00001111b (i.e. 15)- Indicates that the RCS IP Video Call service is enabled for WiFi, 3G, HSPA and LTE cellular access

- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter ID: “rcsIPVideoCallAuth”

Node: /<x>/Services/IR94VideoAuth

Leaf node that represents the authorization for user to use IR94 Video Calling service

Status	Occurrence	Format	Min. Access Types
Optional	One	bool	Get, Replace

Table 114: Services MO sub tree addition parameters (IR94VideoAuth)

- Values: 0, 1
  0- Indicates that IR94 Video Calling service is disabled
  1- Indicates that IR94 Video Calling service is enabled
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter ID: “IR94VideoAuth”

Node: /<x>/Services/allowRCSExtensions

Leaf node that describes whether use of RCS Extensions by the RCS Client is allowed.

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get, Replace

Table 115: Services MO sub tree addition parameters (allowRCSExtensions)

- Values: 0, 1
  0- Not Allowed: The SIP REGISTER shall NOT include the IARIs pertaining to Extensions, and shall NOT include the ICSI for RCS Extension to Extension in its Contact header (default value).
  1- Allowed: The SIP REGISTER shall include IARIs pertaining to Extensions, and the ICSI for RCS Extension to Extension if used by installed applications, in its Contact header.
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter ID: “allowRCSExtensions”

Node: /<x>/Services/Ext

An extension node for Service Provider specific parameters. Clients that are not aware of any extensions in this subtree (e.g. because they are not Service Provider specific) should not instantiate this tree.

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	Node	Get

Table 116: Services MO sub tree addition Service Provider Extension Node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-services:5.2:Ext*
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: “EXT”

A.2.2. IMS sub tree additions

RCS includes the following additions to the IMS MO sub tree where <*IMS*> corresponds to the <*x*> root node of the IMS MO defined in [3GPP TS 24.167].

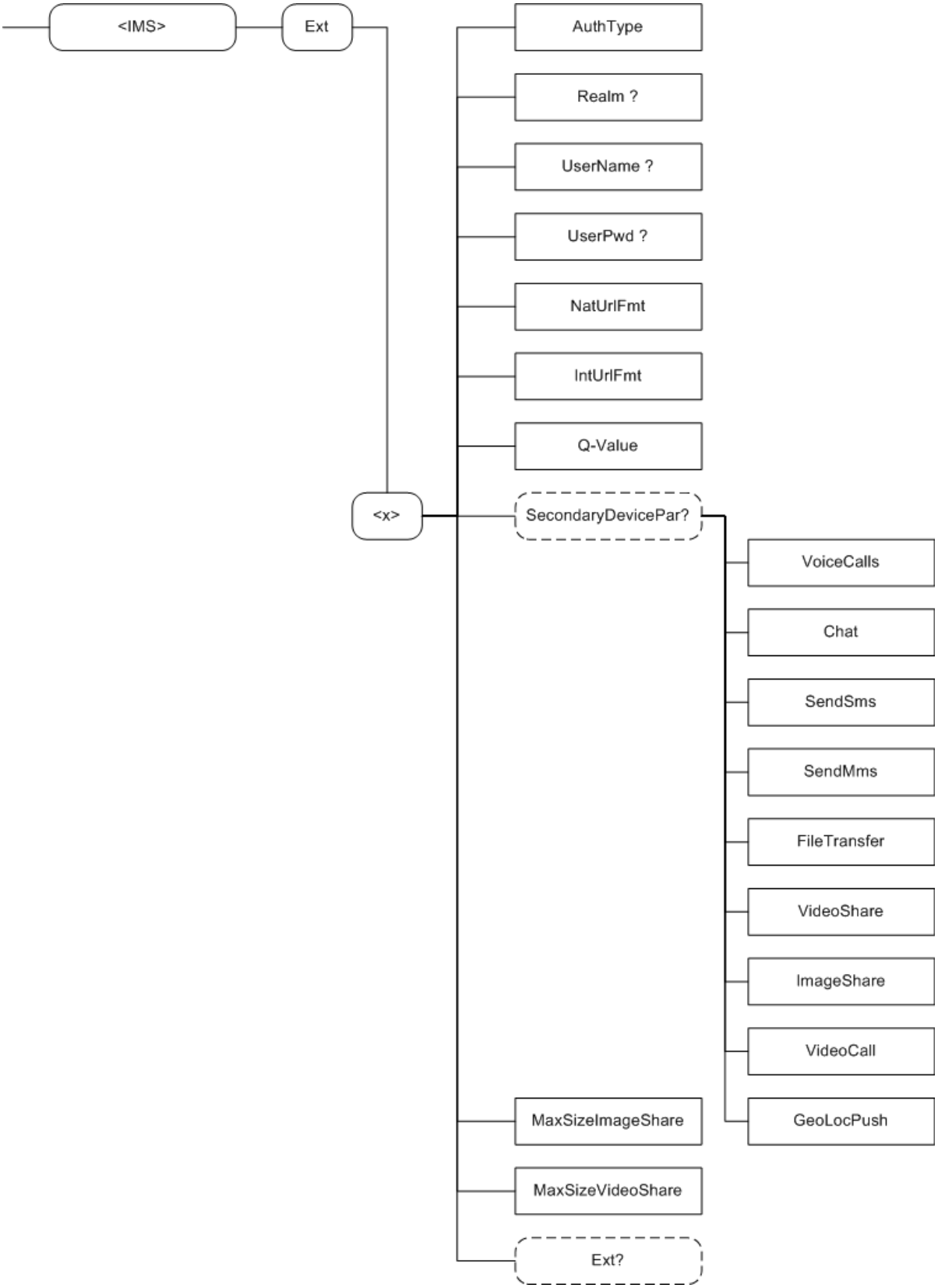


Figure 142: RCS additions to the IMS MO sub tree

The associated HTTP configuration XML structure associated with the IMS parameters (both from the IMS MO defined in [3GPP TS 24.167] and the RCS specific parameters (shown in blue)) is presented in the table below

```
characteristic type="APPLICATION">
        <parm name=”AppID” value=”X”/>
</characteristic>
<characteristic type=”IMS”>
        <parm name=”Name” value=”X”/>
        <characteristic type=”ConRefs”>
              <parm name=”ConRef” value=”X”/>
        </characteristic>
        <parm name=”PDP_ContextOperPref” value=”X”/>
        <parm name=”Timer_T1” value=”X”/>
        <parm name=”Timer_T2” value=”X”/>
        <parm name=”Timer_T4” value=”X”/>
        <parm name=”Private_User_Identity” value=”X”/>
        <characteristic type=”Public_User_Identity_List”>
              <parm name=”Public_User_Identity” value=”X”/>
        </characteristic>
        <parm name=”Home_network_domain_name” value=”X”/>
        <characteristic type=”Ext”>
              <parm name=”NatUrlFmt” value=”X”/>
              <parm name=”IntUrlFmt” value=”X”/>
              <parm name=”Q-Value” value=”X”/>
              <characteristic type=”SecondaryDevicePar”>
                    <parm name=”VoiceCall” value=”X”/>
                    <parm name=”Chat” value=”X”/>
                    <parm name=”SendSms” value=”X”/>
                    <parm name=”FileTransfer” value=”X”/>
                    <parm name=”VideoShare” value=”X”/>
                    <parm name=”ImageShare” value=”X”/>
                    <parm name=”VideoCall” value=”X”/>
                    <parm name=”GeoLocPush” value=”X”/>
              </characteristic>
              <parm name=”MaxSizeImageShare” value=”X”/>
              <parm name=”MaxTimeVideoShare value=”X”/>
              <characteristic type=”Ext”/>
        </characteristic>
        <characteristic type=”ICSI_List”>
              <parm name=”ICSI” value=”X”/>
              <parm name=”ICSI_Resource_Allocation_Mode” value=”X”/>
        </characteristic>
        <characteristic type=”LBO_P-CSCF_Address”>
              <parm name=”Address” value=”X”/>
              <parm name=”AddressType” value=”X”/>
        </characteristic>
        <parm name=”Voice_Domain_Preference_E_UTRAN” value=”X”/>
        <parm name=”SMS_Over_IP_Networks_Indication” value=”X”/>
        <parm name=”Keep_Alive_Enabled” value=”X”/>
        <parm name=”Voice_Domain_Preference_UTRAN” value=”X”/>
        <parm name=”Mobility_Management_IMS_Voice_Termination” value=”X”/>
        <parm name=”RegRetryBaseTime” value=”X”/>
        <parm name=”RegRetryMaxTime” value=”X”/>
        <characteristic type=”PhoneContext_List”>
              <parm name=”PhoneContext” value=”X”/>
              <parm name=”Public_User_Identity” value=”X”/>
        </characteristic>
        <characteristic type=”APPAUTH”>
              <parm name=”AuthType” value=”X”/>
              <parm name=”Realm” value=”X”/>
```

```
            <parm name="UserName" value="X"/>
            <parm name="UserPwd" value="X"/>
        </characteristic>
</characteristic>
```

Table 117 : IMS sub tree associated HTTP configuration XML structure

Node: <x>

Under this interior node the RCS parameters related to IMS are placed

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 118: IMS MO sub tree addition IMS node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-IMS:5.2*
- Associated HTTP XML characteristic type: "IMS"

Node: <x>/AuthType

Leaf node that describes the type of IMS authentication for the user

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Table 119: IMS MO sub tree addition parameters (AuthType)

- Values: 'EarlyIMS', 'AKA', 'Digest'
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: "AuthType"

Node: <x>/Realm

If the IMS mode for authentication is 'digest', this leaf node exists and contains the realm URL affected to the user

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

Table 120: IMS MO sub tree addition parameters (Realm)

- Values: <Realm URL>, example: 'authenticatorY.operatorX.com'
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: "Realm"

Node: <x>/UserName

If the IMS mode for authentication is 'Digest', this leaf node exists and contains the realm User name assigned to the user for IMS authorization/registration

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	No Get, No Copy

Table 121: IMS MO sub tree addition parameters (UserName)

- Values: <use name assigned to user for IMS authentication/registration purpose>, Example: “Alice”
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “UserName”

Node: <x>/UserPwd

If the IMS mode for authentication is ‘Digest’, this leaf node exists and contains the User password assigned to the user for IMS authorization/registration

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	No Get, No Copy

Table 122: IMS MO sub tree addition parameters (UserPwd)

- Values: <password assigned to user for IMS authentication/registration purpose>, Example: ‘secretxyz’
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “UserPwd”

Node: <x>/NatUrlFmt

This leaf node indicates the format (SIP URI or tel URI) to be used when the callee numbering is dialled in national format

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Table 123: IMS MO sub tree addition parameters (NatUrlFmt)

- Values: 0, 1
  0: tel URI format (example: tel:0234578901;phone-context=<home-domain-name>)
  1: SIP URI format (example: sip:0234578901;phone-context=<home-domain-name>@<home-domain-name>;user=phone)
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: “NatUrlFmt”

Node: <x>/IntUrlFmt

This leaf node indicates the format (SIP URI or tel URI) to be used when the callee numbering is dialled in international format

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Table 124: IMS MO sub tree addition parameters (IntUrlFmt)

- Values: 0, 1
  0: tel URI format (example: tel:+32234578901)
  1: SIP URI format (example: sip:+32234578901@<home-domain-name>;user=phone)

- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: “IntUrlFmt”

Node: <x>/QValue

This leaf node indicates the Q-value to be put in the Contact header of the Register method. This can be useful in case of multidevice for the forking algorithm.

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Table 125: IMS MO sub tree addition parameters (QValue)

- Values: ‘0.1’, ‘0.2’, ‘0.3’, ‘0.4’, ‘0.5’, ‘0.6’, ‘0.7’, ‘0.8’, ‘0.9’, ‘1.0’
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “Q-Value”

Node: <x>/SecondaryDevicePar

Presence of this interior node indicates that the RCS device is a secondary device. This node is not instantiated in case of primary device. It is thus only required to be supported on devices/clients that can function as a secondary device.

Under this node are instantiated the parameters necessary to control the ability for the user to restrict RCS services on the secondary device. The notion of primary and secondary device is defined in section 2.9.2.2.

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

Table 126: IMS MO sub tree addition Secondary Device node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-IMS:5.2:SecondaryDevice*
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: “SecondaryDevicePar”

Node: <x>/SecondaryDevicePar/VoiceCalls

This leaf node is instantiated if the device is an RCS secondary device. It allows the Service Provider to authorize or not the device user to control the voice call delivery on this secondary device. The notion of primary and secondary device is defined in section 2.9.2.2.

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Table 127: IMS MO sub tree addition parameters (VoiceCalls)

- Values: 0, 1
  0- Indicates authorization
  1- Indicates non authorization

- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “VoiceCall”

Node: <x>/SecondaryDevicePar/Chat

This leaf node is instantiated if the device is an RCS secondary device. It allows the Service Provider to authorize or not the device user to control the incoming chat session acceptation on this secondary device. The notion of primary and secondary device is defined in section 2.9.2.2.

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Table 128: IMS MO sub tree addition parameters (Chat)

- Values: 0, 1
  0- Indicates authorization
  1- Indicates non authorization
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “Chat”

Node: <x>/SecondaryDevicePar/SendSms

This leaf node is instantiated if the device is an RCS secondary device. It allows the Service Provider to authorize or not the device user to control the restricted SMS service (only possibility to send an SMS on a secondary device) on this secondary device. The notion of primary and secondary device is defined in section 2.9.2.2.

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Table 129: IMS MO sub tree addition parameters (SendSMS)

- Values: 0, 1
  0- Indicates authorization
  1- Indicates non authorization
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “SendSMS”

Node: <x>/SecondaryDevicePar/SendMms

This leaf node is instantiated if the device is an RCS secondary device. It allows the Service Provider to authorize or not the device user to control the restricted MMS service (only possibility to send an MMS on a secondary device) on this secondary device. The notion of primary and secondary device is defined in section 2.9.2.2.

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Table 130: IMS MO sub tree addition parameters (SendMMS)

- Values: 0, 1
  0- Indicates authorization
  1- Indicates non authorization
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “SendMMS”

Node: <x>/SecondaryDevicePar/FileTransfer

This leaf node is instantiated if the device is an RCS secondary device. It allows the Service Provider to authorize or not the device user to control the incoming File Transfer reception on this secondary device. The notion of primary and secondary device is defined in section 2.9.2.2.

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Table 131: IMS MO sub tree addition parameters (FileTransfer)

- Values: 0, 1
  0- Indicates authorization
  1- Indicates non authorization
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “FileTransfer”

Node: <x>/SecondaryDevicePar/VideoShare

This leaf node is instantiated if the device is an RCS secondary device. It allows the Service Provider to authorize or not the device user to control the incoming Video Share session reception on this secondary device. The notion of primary and secondary device is defined in section 2.9.2.2.

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Table 132: IMS MO sub tree addition parameters (VideoShare)

- Values: 0, 1
  0- Indicates authorization
  1- Indicates non authorization
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “VideoShare”

Node: <x>/SecondaryDevicePar/ImageShare

This leaf node is instantiated if the device is an RCS secondary device. It allows the Service Provider to authorize or not the device user to control the incoming Image Share session reception on this secondary device. The notion of primary and secondary device is defined in section 2.9.2.2.

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Table 133: IMS MO sub tree addition parameters (ImageShare)

- Values: 0, 1
  0- Indicates authorization
  1- Indicates non authorization
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “ImageShare”

Node: <x>/SecondaryDevicePar/VideoCall

This leaf node is instantiated if the device is an RCS secondary device. It allows the Service Provider to authorize or not the device user to control the incoming Video Call session reception on this secondary device. The notion of primary and secondary device is defined in section 2.9.2.2.

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Table 134: IMS MO sub tree addition parameters (VideoCall)

- Values: 0, 1
  0- Indicates authorization
  1- Indicates non authorization
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “VideoCall”

Node: <x>/SecondaryDevicePar/GeoLocPush

This leaf node is instantiated if the device is an RCS secondary device. It allows the Service Provider to authorize or not the device user to control the incoming Geolocation PUSH request reception on this secondary device. The notion of primary and secondary device is defined in section 2.9.2.2.

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Table 135: IMS MO sub tree addition parameters (GeoLocPush)

- Values: 0, 1
  0- Indicates authorization
  1- Indicates non authorization
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “GeoLocPush”

Node: <x>/MaxSizeImageShare

Leaf node that represents the maximum authorized size of the content that can be sent in an Image Share session

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

Table 136: IMS MO sub tree addition parameters (MaxSizeImageShare)

- Values: <content maximum size in bytes>. Value equals to 0 means no limitation.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “MaxSizeImageShare”

Node: <x>/MaxTimeVideoShare

Leaf node that represents the maximum authorized duration time for a Video Share session

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

Table 137: IMS MO sub tree addition parameters (MaxTimeVideoShare)

- Values: <Timer value in seconds>. Value equals to 0 means no limitation.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “MaxTimeVideoShare”

Node: <x>/Ext

An extension node for Service Provider specific parameters. Clients that are not aware of any extensions in this subtree (e.g. because they are not Service Provider specific) should not instantiate this tree.

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

Table 138: IMS MO sub tree addition Service Provider Extension Node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-IMS:5.2:Ext*
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: “Ext”

A.2.3. Presence sub tree additions

RCS includes the following additions to the Presence MO sub tree where *<Presence>* corresponds to the *<x>* root node of the Presence MO defined in [PRESENCE2MO].

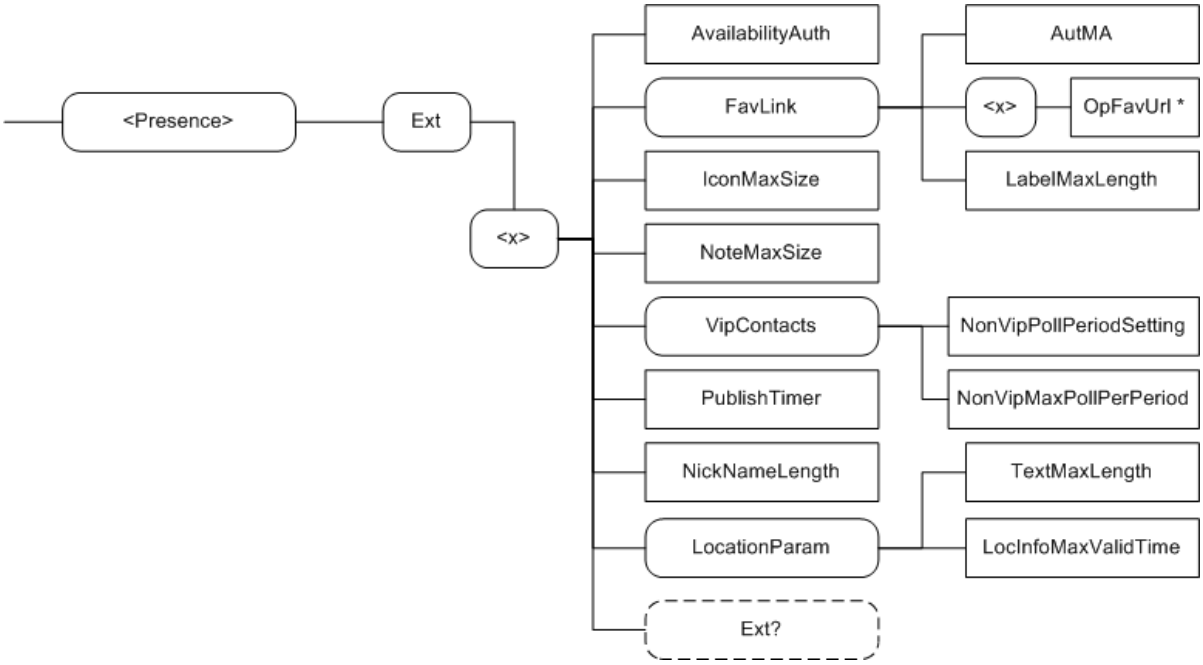


Figure 143: RCS additions to the Presence MO sub tree

The associated HTTP configuration XML structure associated with the Presence parameters (both from the Presence MO defined in [PRESENCE2MO] and the RCS specific parameters (shown in blue)) is presented in the table below

```
<characteristic type="PRESENCE">
        <parm name="AvailabilityAuth" value="X"/>
        <characteristic type="FAVLINK">
        <parm name="AutMa" value="X"/>
              <characteristic type="LINKS">
                    <parm name=" OpFavUrl1" value="X"/>
                    <parm name=" OpFavUrl2" value="X"/>
                    <parm name=" OpFavUrl3" value="X"/>
                    …
              </characteristic>
              <parm name="LabelMaxLength" value="X"/>
        </characteristic>
        <parm name="IconMaxSize" value="X"/>
        <parm name="NoteMaxSize" value="X"/>
        <characteristic type="VIPCONTACTS">
              <parm name="NonVipPollPeriodSetting" value="X"/>
              <parm name="NonVipMaxPollPerPeriod" value="X"/>
        </characteristic>
        <parm name="PublishTimer" value="X"/>
        <parm name="NickNameLength" value="X"/>
        <characteristic type="Location">
              <parm name="TextMaxLength" value="X"/>
              <parm name="LocInfoMaxValidTime" value="X"/>
        </characteristic>
        <characteristic type="Ext"/>
        <parm name="client-obj-datalimit" value="X"/>
        <parm name="content-serveruri" value="X"/>
        <parm name="source-throttlepublish" value="X"/>
        <parm name="max-number-ofsubscriptions-inpresence-list" value="X"/>
        <parm name="service-uritemplate" value="X"/>
        <parm name="RLS-URI" value="X"/>
</characteristic>
```

Table 139 : Presence sub tree associated HTTP configuration XML structure

Node: <x>

Under this interior node the RCS parameters related to Presence are placed

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 140: Presence MO sub tree addition presence node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-Presence:5.2*
- Associated HTTP XML characteristic type: "PRESENCE"

Node: <x>/AvailabilityAuth

Leaf node that describes whether the presence related features are enabled or disabled on the device.

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get

Table 141: Presence MO sub tree addition parameters (AvailabilityAuth)

- Values: 1, the use of Availability status is authorized. 0, the use of Availability status is not authorized.

- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter ID: “AvailabilityAuth”

Node: <x>/FavLink

Interior node under which parameters related to the Service Provider provided Favourite Link(s) are located

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 142: Presence MO sub tree addition Favourite Links node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-Presence:5.2:favlink*
- Associated HTTP XML characteristic type: “FAVLINK”

Node: <x>/FavLink/AutMa

Leaf node that determines the Service Provider policy for Favourite Link instantiation in the local presence document of the presentity

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Table 143: Presence MO sub tree addition parameters (AutMa)

- Values: ‘Auto’, ‘Man’, ‘Auto+Man’.
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter ID: “AutMa”

Node: <x>/FavLink/<x>

A Placeholder interior node where to place 0 or more OpFavUrl leaf nodes

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 144: Presence MO sub tree addition Predefined Links node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-Presence:5.2:favlink:Link-ext*
- Associated HTTP XML characteristic type: “LINKS”

Node: <x>/FavLink/<x>/OpFavUrl

Leaf node that represent a Favourite URL configured by the Service Provider

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	chr	Get

Table 145: Presence MO sub tree addition parameters (OpFavUrl)

- Values: <a Service Provider defined url>

- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “OpFavUrl<X>” where <X> is a positive integer value determining the ordering of the different links

Node: <x>/FavLink/LabelMaxLength

Leaf node that determines the Service Provider policy for Favourite Link instantiation in the local presence document of the presentity

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

Table 146: Presence MO sub tree addition parameters (LabelMaxLength)

- Values: an integer that must be less or equal to 200.

  NOTE: A watcher must be able to display up to 200 characters for this attribute

- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter ID: “LabelMaxLength”

Node: <x>/IconMaxSize

Leaf node that represent the maximum authorized size for an icon

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

Table 147: Presence MO sub tree addition parameters (IconMaxSize)

- Values: <Icon maximum data size in bytes>, the value must be inferior to 204800 (200KB)
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter ID: “IconMaxSize”

Node: <x>/NoteMaxSize

Leaf node that represent the maximum authorized size for a note

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

Table 148: Presence MO sub tree addition parameters (NoteMaxSize)

- Values: < Note maximum length in characters>

  NOTE: This should be set to a value that is lower than the one defined at watcher side in the OMA Presence Implementation guideline [PRESENCEIG].

- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter ID: “NoteMaxSize”

Node: <x>/PublishTimer

Leaf node that indicates the timer value for the Presence Publish refreshment

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

Table 149: Presence MO sub tree addition parameters (PublishTimer)

- Values: < Timer value in seconds>
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “PublishTimer”

Node: <x>/NickNameLength

Leaf node that represents the maximum number of characters allowed for the user chosen nickname.

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

Table 150: Presence MO sub tree addition parameters (NickNameLength)

- Values: must be less or equal to 200

  NOTE: An RCS client must be able to handle of up to 200 characters

- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “NickNameLength”

Node: <x>/LocationParam

Interior node where Location related parameters are stored

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 151: Presence MO sub tree addition Location Parameters node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-Presence:5.2:Location*
- Associated HTTP XML characteristic type: “Location”

Node: <x>/LocationParam/TextMaxLength

Leaf node that represents the maximum numbers of characters authorized for the textual attribute of the Location information

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

Table 152: Presence MO sub tree addition parameters (TextMaxLength)

- Values: must be less or equal to 200.

  NOTE: A watcher must be able to render of up to 200 characters

- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “TextMaxLength”

Node: <x>/LocationParam/LocInfoMaxValidTime

Leaf node that represents the maximum validity duration time for a location item.

This parameter must be taken account by the device presence UA when setting the “until” attribute of the presence items place-type, time-offset and the usage-rule/retention-expiry item value

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

Table 153: Presence MO sub tree addition parameters (LocInfoMaxValidTime)

- Values: < Validity time in seconds>, when set to 0 there is no limit to the validity time
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “LocInfoMaxValidTime”

Node: <x>/VipContacts

Interior node where VIP contacts related parameters are stored

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 154: Presence MO sub tree addition VIP Contacts node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-Presence:5.2:VipContacts*
- Associated HTTP XML characteristic type: “VIPCONTACTS”

Node: <x>/VipContacts/NonVipPollPeriodSetting

Leaf node that indicates, in seconds, the period duration for the calculation of the number of Poll operations on the non-VIP Contacts (“rcs_poll”) RLS list authorized during this period

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

Table 155: Presence MO sub tree addition parameters (NonVipPollPeriodSetting)

- Values: integer that represents a time value in seconds
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “NonVipPollPeriodSetting”

Node: <x>/VipContacts/NonVipMaxPollPerPeriod

Leaf node that indicates the maximum number of Poll operations on the non-VIP Contacts (“rcs_poll”) RLS list that are authorized for the User Agent during each period (period parameter defined in the previous /VipContacts/NonVipPollPeriodSetting node).

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

Table 156: Presence MO sub tree addition parameters (NonVipMaxPollPerPeriod)

- Values: integer that represents the total amount of Poll operations on the non-VIP Contacts list per each period.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “NonVipMaxPollPerPeriod”

Node: <x>/Ext

An extension node for Service Provider specific parameters. Clients that are not aware of any extensions in this subtree (e.g. because they are not Service Provider specific) should not instantiate this tree.

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

Table 157: Presence MO sub tree addition Service Provider Extension Node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-Presence:5.2:Ext*
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: “Ext”

A.2.4. XDMS sub tree additions

RCS includes the following additions to the XDMS MO sub tree where *<XDMS>* corresponds to the *<x>* root node of the XDMS MO defined in [XDMMO].

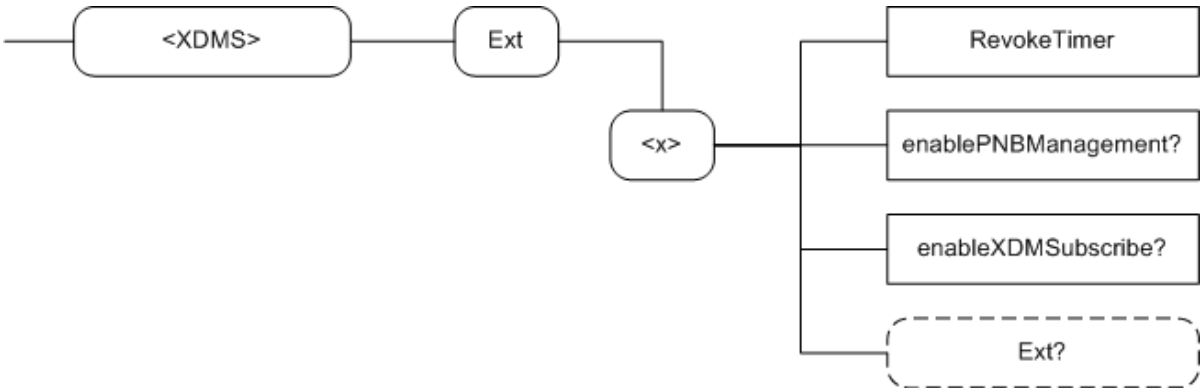


Figure 144: RCS additions to the XDMS MO sub tree

The associated HTTP configuration XML structure associated with the XDMS parameters (both from the XDMS MO defined in [XDMMO] and the RCS specific parameters (shown in blue)) is presented in the table below

```
<characteristic type="XDMS">
        <parm name="RevokeTimer" value="X"/>
        <parm name="enablePNBManagement" value="X"/>
        <parm name="enableXDMSubscribe" value="X"/>
        <characteristic type="Ext"/>
        <parm name="XCAPRootURI" value="X"/>
        <parm name="XCAPAuthenticationUserName" value="X"/>
        <parm name="XCAPAuthenticationSecret" value="X"/>
        <parm name="XCAPAuthenticationType" value="X"/>
</characteristic>
```

Table 158 : XDMS sub tree associated HTTP configuration XML structure

Node: <x>

Under this interior node the RCS parameters related to XDM are placed

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 159: XDM MO sub tree addition xdm node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-xdm:5.2*
- Associated HTTP XML characteristic type: "XDMS"

Node: <x>/RevokeTimer

Leaf node that indicates the duration a contact should remain in the RCS revocation list. It may also be used for the frequency that the list is checked.

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

Table 160: XDMS MO sub tree addition parameters (RevokeTimer)

- Values: < Timer value in seconds>
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: "RevokeTimer"

Node: <x>/enablePNBManagement

Leaf node that describes whether the PNB feature is turned ON or OFF

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get, Replace

Table 161: XDMS MO sub tree addition parameters (enablePNBManagement)

- Values:
  0 or not instantiated, the PNB feature is not used.
  1, the PNB feature is ON, the PNB lists can be managed by the user and applied by BPEF
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: "enablePNBManagement"

Node: <x>/enableXDMSubscribe

Leaf node that describes whether the client may subscribe to XDM Document Changes

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get, Replace

Table 162: XDMS MO sub tree addition parameters (enableXDMSubscribe)

- Values:
  0 or not instantiated, the client shall not subscribe to XDM Document Changes and fetch the latest version whenever the user wants to modify them.
  1, the client shall cache the XDM Documents and subscribe for changes as described in section 2.14.2.1
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “enableXDMSubscribe”

Node: <x>/Ext

An extension node for Service Provider specific parameters. Clients that are not aware of any extensions in this subtree (e.g. because they are not Service Provider specific) should not instantiate this tree.

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

Table 163: XDMS MO sub tree addition Service Provider Extension Node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-xdm:5.2:Ext*
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: “Ext”

A.2.5. SUPL sub tree additions

RCS includes the following additions to the SUPL MO sub tree where <*SUPL*> corresponds to the <*x*> root node of the SUPL MO defined in [SUPLMO]:

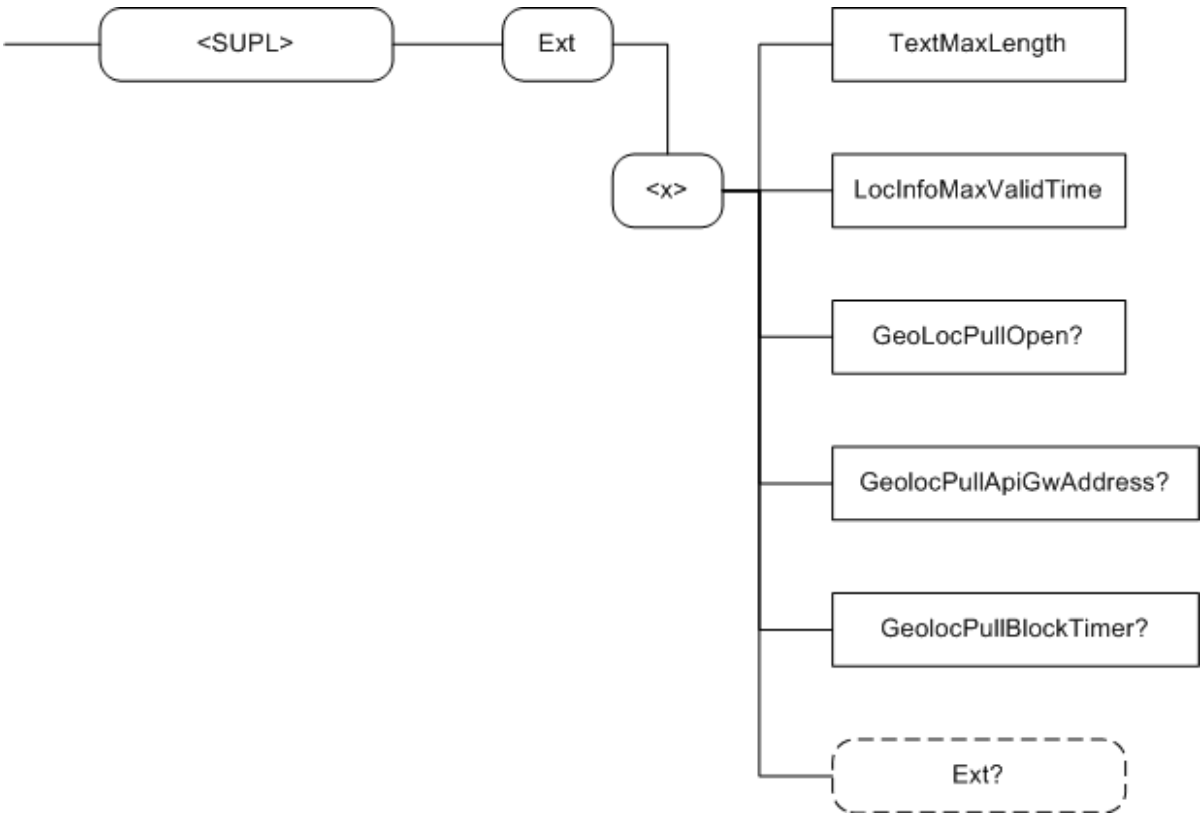


Figure 145 : RCS additions to the SUPL MO sub tree

The associated HTTP configuration XML structure associated to the geolocation parameters (both from the SUPL MO defined in [SUPLMO] and the RCS specific parameters (shown in blue)) is presented in the table below

```
<characteristic type="SUPL">
        <parm name="TextMaxLength" value="X"/>
        <parm name="LocInfoMaxValidTime" value="X"/>
        <parm name="geolocPullOpen" value="X"/>
        <parm name="geolocPullApiGwAddress" value="X"/>
        <parm name="geolocPullBlockTimer" value="X"/>
        <characteristic type="Ext"/>
        <parm name="Addr" value="X"/>
        <parm name="AddrType" value="X"/>
</characteristic>
```

Table 164 : SUPL sub tree associated HTTP configuration XML structure

Node: <x>

Under this interior node the RCS parameters related to the geolocation configuration are placed.

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 165: SUPL MO sub tree addition geoloc node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-supl:5.2*
- Associated HTTP XML characteristic type: "SUPL"

Node: <x>/TextMaxLength

Leaf node that represents the maximum numbers of characters authorized for the textual attribute of the location information provided in the geolocation PUSH and PULL services

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

Table 166: SUPL MO sub tree addition parameters (TextMaxLength)

- Values: must be less or equal to 200.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “TextMaxLength”

Node: <x>/LocInfoMaxValidTime

Leaf node that represents the maximum validity duration time for a location item

This parameter must be taken account by the device providing the location information when setting the “until” attribute of the items time-offset and the usage-rule/retention-expiry item value

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

Table 167: SUPL MO sub tree addition parameters (LocInfoMaxValidTime)

- Values: < Validity time in seconds>
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “LocInfoMaxValidTime”

Node: <x>/GeoLocPullOpen

Leaf node that represents the service provider policy concerning the possibility to retrieve the location of non-RCS users

The parameter is only applicable in case the Geolocation PULL service based on LBS is supported. It will not be instantiated otherwise.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get, Replace

Table 168: SUPL MO sub tree addition parameters (GeoLocPullOpen)

- Values: 0, 1
  0- Indicates that Geolocation PULL service is only authorised if the target user is an RCS user
  1- Indicates that Geolocation PULL service is authorised if the target user is an RCS user or not

  NOTE: If the leaf node is not instantiated, by default, the service is restricted to RCS target users

- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “geolocPullOpen”

Node: <x>/GeolocPullApiGwAddress

Leaf node that represents the URL address of the GeoLocationPull API Gateway

The parameter is only applicable in case the Geolocation PULL service based on LBS is supported. It will not be instantiated otherwise.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get, Replace

Table 169: SUPL MO sub tree addition parameters (GeolocPullApiGwAddress)

- Values: URL address of the Pull API gateway
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “geolocPullApiGwAddress”

Node: <x>/GeolocPullBlockTimer

Leaf node that represents the interval during which the Geolocation PULL application is not allowed to send a PULL request to a target contact if a previous request was explicitly rejected. The parameter is only applicable if the Geolocation PULL service is based on File Transfer technology and will thus not be instantiated in case that is not supported.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get, Replace

Table 170: SUPL MO sub tree addition parameters (GeolocPullBlockTimer)

- Values: <Timer value in seconds>
  The value represents the duration, in case of explicit operation denied by a target contact, during which a new Pull operation in not allowed to be initiated.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “geolocPullBlockTimer”

Node: <x>/Ext

An extension node for Service Provider specific parameters. Clients that are not aware of any extensions in this subtree (e.g. because they are not Service Provider specific) should not instantiate this tree.

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

Table 171: SUPL MO sub tree addition Service Provider Extension Node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-supl:5.2:Ext*
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: “Ext”

A.2.6. IM sub tree additions

RCS includes the following additions to the IM MO sub tree where *<IM>* corresponds to the *<x>* root node of the IM MO described in [RCS5-SIMPLEIM-ENDORS]:

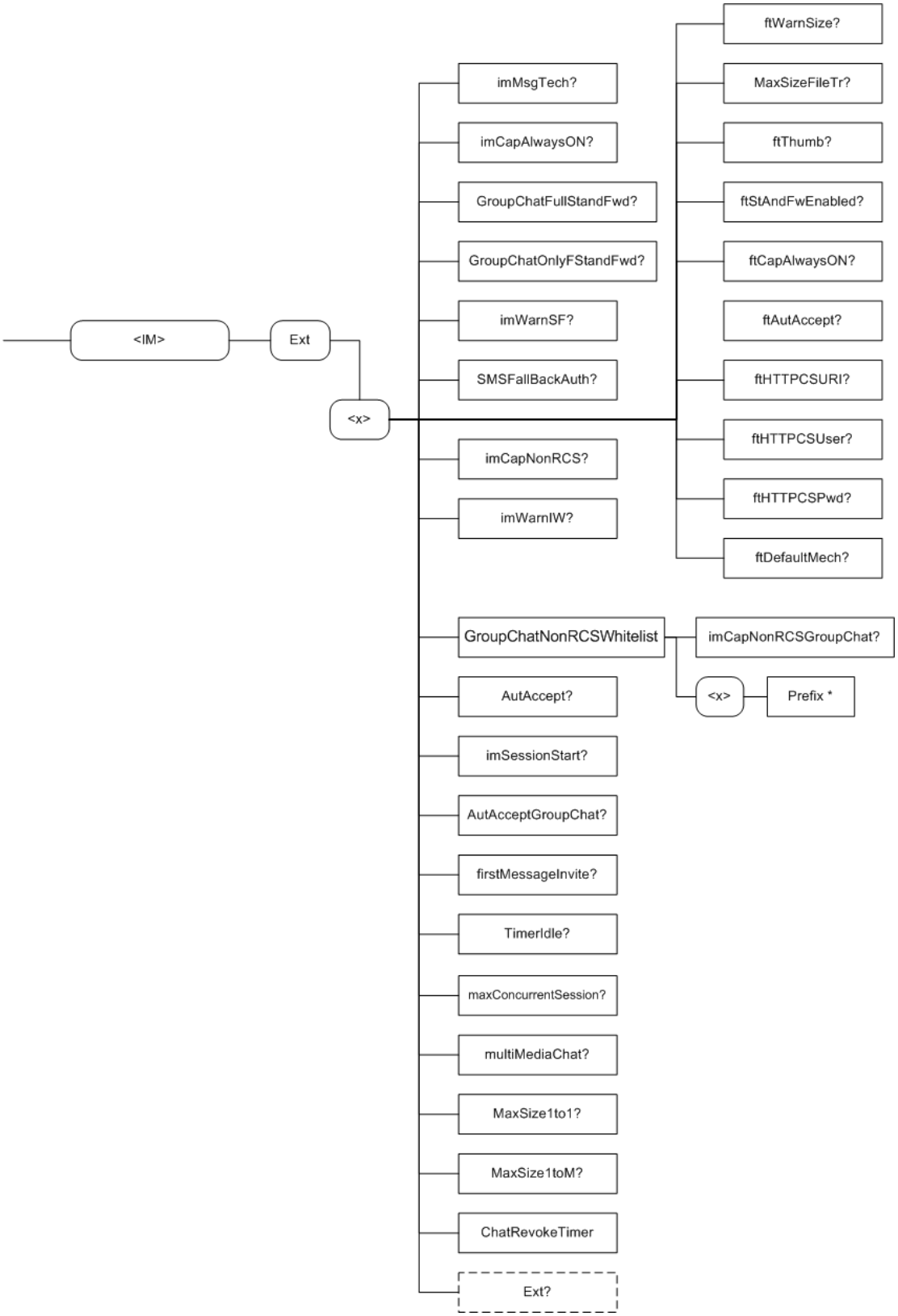


Figure 146: RCS additions to the IM MO sub tree

The associated HTTP configuration XML structure associated to the IM parameters (both from the IM MO defined in [RCS5-SIMPLEIM-ENDORS] and the RCS specific parameters (shown in blue)) is presented in the table below

```
<characteristic type="IM">
        <parm name="imMsgTech" value="X"/>
        <parm name="imCapAlwaysON" value="X"/>
        <parm name="GroupChatFullStandFwd" value="X"/>
        <parm name="GroupChatOnlyFStandFwd" value="X"/>
        <parm name="imWarnSF" value="X"/>
        <parm name="SmsFallBackAuth" value="X"/>
        <parm name="imCapNonRCS" value="X"/>
        <parm name="imWarnIW" value="X"/>
        <characteristic type="GroupChatNonRCSWhitelist" value="X"/>
              <parm name="imCapNonRCSGroupChat" value="X"/>
              <characteristic type="GroupChatAllowedPrefixes">
                    <parm name=" Prefix1" value="X"/>
                    <parm name=" Prefix2" value="X"/>
                    <parm name=" Prefix3" value="X"/>
                    …
              </characteristic>
        </characteristic>
        <parm name="AutAccept" value="X"/>
        <parm name="AutAcceptGroupChat" value="X"/>
        <parm name="imSessionStart" value="X"/>
        <parm name="firstMessageInvite" value="X"/>
        <parm name="TimerIdle" value="X"/>
        <parm name="MaxConcurrentSession" value="X"/>
        <parm name="multiMediaChat" value="X"/>
        <parm name="MaxSize1to1" value="X"/>
        <parm name="MaxSize1toM" value="X"/>
        <parm name="ChatRevokeTimer" value="X"/>
        <parm name="ftWarnSize" value="X"/>
        <parm name="MaxSizeFileTr" value="X"/>
        <parm name="ftThumb" value="X"/>
        <parm name="ftStAndFwEnabled" value="X"/>
        <parm name="ftCapAlwaysON" value="X"/>
        <parm name="ftAutAccept" value="X"/>
        <parm name="ftHTTPCSURI" value="X"/>
        <parm name="ftHTTPCSUser" value="X"/>
        <parm name="ftHTTPCSPwd" value="X"/>
        <parm name="ftDefaultMech" value="X"/>
        <characteristic type="Ext"/>
        <parm name="pres-srv-cap" value="X"/>
        <parm name="deferred-msg-func-uri" value="X"/>
        <parm name="max_adhoc_group_size" value="X"/>
        <parm name="conf-fcty-uri" value="X"/>
        <parm name="exploder-uri" value="X"/>
</characteristic>
```

Table 172 : IM sub tree associated HTTP configuration XML structure

Node: <x>

Under this interior node the RCS parameters related to the IM configuration are placed.

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 173: IM MO sub tree addition IM node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-im:5.2*
- Associated HTTP XML characteristic type: “IM”

Node: <x>/imMsgTech

Leaf node that describes parameter allows selecting what technology is used for the chat service described in sections 3.3 and 3.4 as well as for the File Transfer service in section 3.5.

It is required to be instantiated if a service provider enables Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get, Replace

Table 174: IM MO sub tree addition parameters (imMsgTech)

- Values: 1, CPM as specified in [RCS5-CPM-CONVFUNC-ENDORS]. 0 (default if not provided), SIMPLE IM as specified in [RCS5-SIMPLEIM-ENDORS].
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML parameter ID: “imMsgTech”

Node: <x>/imCapAlwaysON

Leaf node that describes whether the Chat capability needs to be on independently of whether or not the other end is registered. For example this can be used in Service Providers providing the store and forward functionality for Chat

It is required to be instantiated if a service provider enables Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get, Replace

Table 175: IM MO sub tree addition parameters (IMCAPAlwaysOn)

- Values: 1, RCS Messaging Server based store and forward is enabled; 0, it is disabled
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “imCapAlwaysOn”

Node: <x>/imWarnSF

Leaf node that describes whether the UX should alert the user that messages are handled differently when the store and forward functionality is involved.

It is required to be instantiated if a service provider enables Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get, Replace

Table 176: IM MO sub tree addition parameters (imWarnSF)

- Values: 1, the user is made aware via the UX when the messages are deferred using S&F. 0, the user is not aware that messages are deferred.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.

- Associated HTTP XML parameter ID: “imWarnSF”

Node: <x>/GroupChatFullStandFwd

Leaf node that represents whether the service provider for the device provides the full store and forward feature for Group Chat

It is required to be instantiated if a service provider enables Group Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get, Replace

Table 177: IM MO sub tree addition parameters (GroupChatFullStandFwd)

- Values: 0, 1
  0- Indicates no support for Full Store and Forward for Group Chat (default value)
  1- Indicates support for Full Store and Forward for Group Chat
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “GroupChatFullStandFwd”

Node: <x>/GroupChatOnlyFStandFwd

Leaf node that represents whether the service provider allows all users to be invited for a group chat or only those that support the full store and forward feature for Group Chat

It is only relevant to be instantiated if a service provider enables Group Chat, but is not required to be instantiated even then.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get, Replace

Table 178: IM MO sub tree addition parameters (GroupChatOnlyFStandFwd)

- Values: 0, 1
  0 or not provided- Indicates all users may be invited for Group Chat regardless of their support for Full Store and Forward
  1- Indicates that only users that support for Full Store and Forward may be invited for Group Chat
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “GroupChatOnlyFStandFwd”

Node: <x>/SmsFallbackAuth

Leaf node that represents the authorization for the device to propose automatically a SMS fallback in case of chat initiation failure

It is required to be instantiated if a service provider enables 1-to-1 Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get

Table 179: IM MO sub tree addition parameters (SmsFallbackAuth)

- Values: 0, 1
  0- Indicates authorization is not granted, i.e. fallback is disabled
  1- Indicates authorization is granted, i.e. fallback is enabled

- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “SmsFallbackAuth”

Node: <x>/imCapNonRCS

Leaf node that describes whether the Chat capability needs to be on independently of whether or not the other end is an RCS contact. For example this can be used in Service Providers providing the interworking functionality for Chat

It is required to be instantiated if a service provider enables Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get, Replace

Table 180: IM MO sub tree addition parameters (IMCAPNonRCS)

- Values: 1, RCS Messaging Server based interworking is enabled; 0, it is disabled
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “imCapNonRCS”

Node: <x>/imWarnIW

Leaf node that describes whether the UX should alert the user that messages are handled differently when the interworking functionality is involved

It is required to be instantiated if a service provider enables Chat

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get, Replace

Table 181: IM MO sub tree addition parameters (IMWarnIW)

- Values: 1, the user is made aware via the UX when the messages are interworked to SMS/MMS. 0, the user is not aware that messages are interworked.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “imWarnIW”

Node: <x>/GroupChatNonRCSWhitelist

A Placeholder interior node for the configuration related to a whitelist for non RCS users invited to a group chat

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 182: IM MO sub tree GroupChatNonRCSWhitelist node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-im:5.2:gcnonrcswhitelist*
- Associated HTTP XML parameter ID: “GroupChatNonRCSWhitelist”

Node: <x>/GroupChatNonRCSWhitelist/imCapNonRCSGroupChat

Leaf node that describes whether and under which conditions the device is able to add non RCS contacts in a Group Chat. For example this can be used by Service Providers providing the interworking functionality for Group Chat.

It is required to be instantiated if a service provider enables Group Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get, Replace

Table 183: IM MO sub tree addition parameters (IMCAPNonRCSGroupChat)

- Values: This parameter can have 3 possible values:
  0 - The device is not able to add any non RCS contact in any Group Chat session
  1 - The device is able to add non RCS contacts in any Group Chat session
  2 - The device is able to add non RCS contacts only upon Group Chat creation and only in Group Chat sessions generated by its user
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “imCapNonRCSGroupChat”

Node: <x>/GroupChatNonRCSWhitelist/<x>

A Placeholder interior node where to place 1 or more Prefix leaf nodes

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 184: IM MO sub tree GroupChatAllowedPrefixes node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-im:5.2:gcnonrcswhitelist:prefixes*
- Associated HTTP XML characteristic type: “GroupChatAllowedPrefixes”

Node: <x>/GroupChatNonRCSWhitelist/<x>/Prefix

Leaf node that represents a prefix configured by the Service Provider

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	chr	Get, Replace

Table 185: IM MO sub tree addition parameters (Prefix)

- Values: <a Service Provider defined prefix>
  The prefix is used to identify phone numbers of non RCS contacts allowed to be added to a Group Chat. The phone numbers are identified by matching with the prefix value starting from the left. Its length can be one or more digits and it can start with the "+" character.
  Examples:, +446, +4479, 00446, 004479, 06, 079
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “Prefix<X>” where <X> is a positive integer value

Node: <x>/AutAccept

Leaf node that represent the automatic/manual chat session answer mode

It is required to be instantiated if a service provider enables 1-to-1 Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get

Table 186: IM MO sub tree addition parameters (AutAccept)

- Values: 0, 1
  0- Indicates manual answer mode
  1- Indicates automatic answer mode (default value)
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “AutAccept”

Node: <x>/imSessionStart

Leaf node that describes when the receiver client/device implementation should return the 200 OK initiating the MSRP session associated to a 1-to-1 chat. Please note that this parameter is transparent to the user.

It is required to be instantiated if a service provider enables 1-to-1 Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get, Replace

Table 187: IM MO sub tree addition parameters (imSessionStart)

- Values: This parameter can have 3 possible values:
  - 0 (RCS 5.1 default):
    The 200 OK is sent when the receiver consumes the notification by opening the chat window.
  - 1 (RCS Release 2-4 default):
    The 200 OK is sent when the receiver starts to type a message to be sent back in the chat window.
  - 2 (new option):
    The 200 OK is sent when the receiver presses the button to send a message (That is the message will be buffered in the client until the MSRP session is established).
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “imSessionStart”

Node: <x>/AutAcceptGroupChat

Leaf node that represent the automatic/manual Group Chat session answer mode

It is required to be instantiated if a service provider enables Group Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get

Table 188: IM MO sub tree addition parameters (AutAcceptGroupChat)

- Values: 0, 1
  0- Indicates manual answer mode
  1- Indicates automatic answer mode (default value)

- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “AutAcceptGroupChat”

Node: <x>/firstMsgInvite

Leaf node that controls whether the initial message in the chat is sent in a CPIM body of the SIP INVITE request or can only be sent once the MSRP session has been set up

It is required to be instantiated if a service provider enables 1-to-1 Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get, Replace

Table 189: IM MO sub tree addition parameters (firstMsgInvite)

- Values: 0, the message is sent in the MSRP, 1, the message is added to the INVITE request as a CPIM body
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “firstMessageInvite”

Node: <x>/TimerIdle

Leaf node that represents the timeout for a chat session in idle mode (when there is no chat user activity)

It is required to be instantiated if a service provider enables Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

Table 190: IM MO sub tree addition parameters (TimerIdle)

- Values: <Timer value in seconds>
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “TimerIdle”

Node: <x>/MaxConcurrentSession

Leaf node that represent the maximum authorized number of Chat (1-to-1 or Group Chat) sessions established from the device. Once this number is reached a new Chat session may not be established anymore until another session is torn down.

It is required to be instantiated if a service provider enables Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

Table 191: IM MO sub tree addition parameters (MaxConcurrentSession)

- Values: <max number of concurrent sessions>, when set to 0 this limit does not apply
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “MaxConcurrentSession”

Node: <x>/multiMediaChat

Leaf node that controls whether or not the device can send and receive other content than text in the chat session

It is not required to be instantiated if a service provider does not enable Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get

Table 192: IM MO sub tree addition parameters (multiMediaChat)

- Values:
  0 (or not provided), the device can only sent and receive text content within the chat . The client should handle the SDP negotiation accordingly
  1, all content allowed by [RCS5-SIMPLEIM-ENDORS] or [RCS5-CPM-CONVFUNC-ENDORS] may be sent in the chat session
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “multiMediaChat”

Node: <x>/MaxSize1To1

Leaf node that represent the maximum authorized size of the content payload of a chat message in a 1 To 1 chat session without transfer encoding. The parameter is applicable for 1-to-1 Chat messages transferred either via SIP INVITE or MSRP requests.

It is required to be instantiated if a service provider enables 1-to-1 Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

Table 193: IM MO sub tree addition parameters (MaxSize1To1)

- Values: <content maximum size in bytes>
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “MaxSize1To1”

Node: <x>/MaxSize1ToM

Leaf node that represent the maximum authorized size of the content payload of a chat message in a Group Chat session without transfer encoding.

It is required to be instantiated if a service provider enables Group Chat.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

Table 194: IM MO sub tree addition parameters (MaxSize1ToM)

- Values: <content maximum size in bytes>
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “MaxSize1ToM”

Node: <x>/ChatRevokeTimer

Leaf node that represents the time the service provider allows to elapse after the client has sent the message and before Revoke Message request is automatically triggered by the client when it has not received the delivery notification for that message.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

Table 195: IM MO sub tree addition parameters (ChatRevokeTimer)

- Values: <Timer value in seconds>
  When set to 0, the client is not able to send MessageRevoke requests
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “ChatRevokeTimer”

Node: <x>/ftWarnSize

Leaf node that describes the file transfer size threshold (in KB) when the user should be warned about the potential charges associated to the transfer of a large file.

It is required to be instantiated if a service provider enables File Transfer.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get, Replace

Table 196: IM MO sub tree addition parameters (ftWarnSize)

- Values: The file size threshold (in KB) or 0 to disable the warning
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “ftWarnSize”

Node: <x>/MaxSizeFileTr

Leaf node that represent the maximum authorized size of a file that can be transfers using the RCS File Transfer service

It is required to be instantiated in case a service provider enables File Transfer.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

Table 197: IM MO sub tree addition parameters (MaxSizeFileTr)

- Values: The maximum file size threshold (in KB) or 0 to disable the limit
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “MaxSizeFileTr”

Node: <x>/ftThumb

Leaf node that controls whether or not the device can send and receive a thumbnail in a File Transfer over MSRP invitation

It is required to be instantiated if a service provider enables File Transfer.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get

Table 198: IM MO sub tree addition parameters (ftThumb)

- Values: 0 (or not provided), the device cannot send or receive a thumbnail in the File Transfer over MSRP invitation and should handle the capability exchange accordingly. 1, the device may send or receive thumbnails in the File Transfer over MSRP Invitation
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “ftThumb”

Node: <x>/ftStAndFwEnabled

Leaf node that describes whether the File Transfer store and forward functionality is enabled.

It is required to be instantiated if a service provider enables File Transfer.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get, Replace

Table 199: IM MO sub tree addition parameters (ftStAndFwEnabled)

- Values: 1, the file store and forward functionality is enabled and, consequently, the file transfer store and forward capability is reported as available. 0, means the opposite and, consequently, the capability is reported as not available.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “ftStAndFwEnabled”

Node: <x>/ftCapAlwaysON

Leaf node that describes whether the file transfer capability needs to be on independently of whether or not the other end is registered

It is required to be instantiated if a service provider enables File Transfer.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get, Replace

Table 200: IM MO sub tree addition parameters (ftCapAlwaysOn)

- Values: 1, RCS Messaging Server based store and forward is enabled; 0, it is disabled. It shall be taken into account that this parameter can be only set to 1 if:
  - All the interconnected service providers support the file transfer store and forward feature, or,
  - store and forward for files is provided as an originating function
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “ftCapAlwaysOn”

Node: <x>/ftAutAccept

Leaf node that describes whether a File Transfer invitation can be automatically accepted

It is required to be instantiated if a service provider enables File Transfer.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get, Replace

Table 201: IM MO sub tree addition parameters (ftAutAccept)

- Values:
  0, automatic acceptance is not possible (regardless of the size of the file).
  1, the File Transfer invitation shall be accepted if the size of the file is smaller than the File Transfer warning size as configured by the FT WARN SIZE parameter
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “ftAutAccept”

Node: <x>/ftHTTPCSURI

This parameter configures the URI of the HTTP content server where files are going to be uploaded on the originating side if the destination cannot accept within the validity period.

NOTE: It is not required to be instantiated because it is not mandatory for a service provider to have this originating solution based on a HTTP content server:

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get, Replace

Table 202: IM MO sub tree addition parameters (ftHTTPCSURI)

- Values: The string containing the URI of the HTTP content server in charge of storing the files.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “ftHTTPCSURI”

Node: <x>/ftHTTPCSUser

This parameter is the value of the user value that shall be used to authenticate the RCS client trying to either get a root URL (HTTP GET request) or upload a file (HTTP post request). Again, note it is not required to be instantiated because it is not mandatory for a service provider to have this originating solution based on a HTTP content server:

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	No Get, No Copy

Table 203: IM MO sub tree addition parameters (ftHTTPCSUser)

- Values: The string containing **user value**.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “ftHTTPCSUser”

Node: <x>/ftHTTPCSPwd

This parameter is the value of the password value that shall be used to authenticate the RCS client trying to either get a root URL (HTTP GET request) or upload a file (HTTP post

request). Again, note it is not required to be instantiated because it is not mandatory for a service provider to have this originating solution based on a HTTP content server:

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	No Get, No Copy

Table 204: IM MO sub tree addition parameters (ftHTTPCSPwd)

- Values: The string containing **password value**.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “ftHTTPCSPwd”

Node: <x>/ftDefaultMech

Leaf node that describes which file transfer mechanism (MSRP or HTTP) shall be used by default if both ends support both of them

It is required to be instantiated if a Service Provider enables File Transfer and configures the URI for the HTTP content server to enable File Transfer using HTTP.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	char	Get, Replace

Table 205: IM MO sub tree addition parameters (ftDefaultMech)

- Values:
  - MSRP
  - HTTP
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “ftDefaultMech”

Node: <x>/Ext

An extension node for Service Provider specific parameters. Clients that are not aware of any extensions in this subtree (e.g. because they are not Service Provider specific) should not instantiate this tree.

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

Table 206: IM MO sub tree addition Service Provider Extension Node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-im:5.2:Ext*
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: “Ext”

A.2.7. CPM MO sub tree

RCS includes the following additions as a new configuration sub tree, the CPM MO subtree

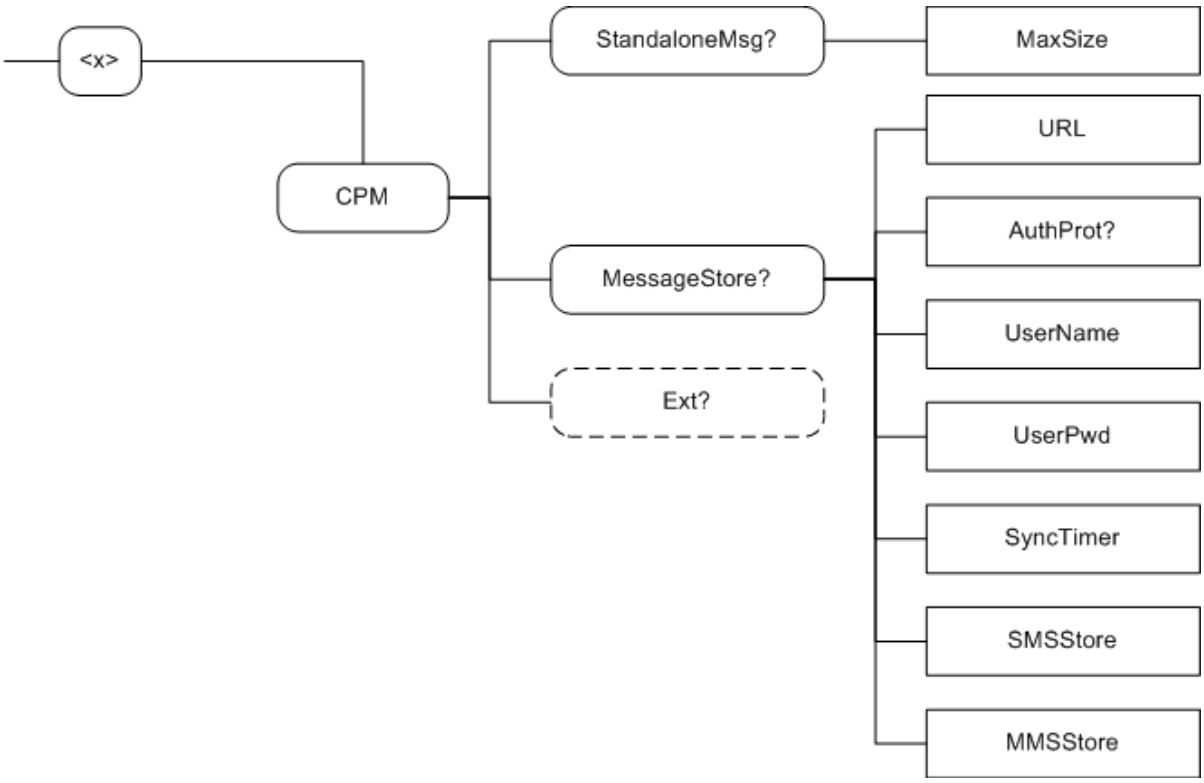


Figure 147: RCS additions, CPM MO sub tree

The associated HTTP configuration XML structure associated to the CPM parameters is presented in the table below. Only RCS specific parameters (shown in blue) are included as OMA does not define a CPM MO.

```
<characteristic type="CPM">
        <characteristic type="StandaloneMsg">
                <parm name="MaxSizeStandalone" value="X"/>
        </characteristic>
        <characteristic type="MessageStore">
                <parm name="Url" value="X"/>
                <parm name="AuthProt" value="X"/>
                <parm name="UserName" value="X"/>
                <parm name="UserPwd" value="X"/>
                <parm name="SyncTimer" value="X"/>
                <parm name="SMSStore" value="X"/>
                <parm name="MMSStore" value="X"/>
        </characteristic>
        <characteristic type="Ext"/>
</characteristic>
```

Table 207 : CPM sub tree associated HTTP configuration XML structure

Node: /<x>/CPM

Under this interior node the RCS parameters related to the CPM configuration are placed.

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 208: CPM MO sub tree addition CPM node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-cpm:5.2*
- Associated HTTP XML characteristic type: “CPM”

Node: /<x>/CPM/StandaloneMsg

Interior node where are filled parameters related to the RCS Text message and Multimedia message service

This node is not instantiated if the Service Provider does not enable Standalone Messaging.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

Table 209: CPM MO sub tree addition Standalone messaging node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-cpm:5.2:StandaloneMsg*
- Associated HTTP XML characteristic type: “StandaloneMsg”

Node: /<x>/CPM/StandaloneMsg/MaxSize

Leaf node that represents the maximum authorized size of the content payload of a text or multimedia message without transfer encoding

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

Table 210: CPM MO sub tree addition parameters (MaxSize)

- Values: <content maximum size in bytes>
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “MaxSize”

Node: /<x>/CPM/MessageStore

Interior node where there are filled parameters related to RCS CPM Common Message Store

This node is not instantiated if the Service Provider does not provide the Common Message Store.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

Table 211: CPM MO sub tree addition Message Store node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-cpm:5.2:MessageStore*
- Associated HTTP XML characteristic type: “MessageStore”

Node: /<x>/CPM/MessageStore/Url

Leaf node that represents the URL address of the Message Store Server

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Table 212: CPM MO sub tree addition parameters (Url)

- Values: the URL for accessing the Message Store Server, if set to an empty string, the Common Message Store is not available.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “Url”

Node: /<x>/CPM/MessageStore/AuthProt

Optional leaf node that can be used to force the Message Store Client to use one of the 2 authentication methods defined in [RCS5-CPM-MSGSTOR-ENDORS]. If not instantiated, but the /URL node in Table 212 is instantiated or present, the Message Store Client SHALL assume the same method as if value 0 had been specified.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get

Table 213: CPM MO sub tree addition parameters (AuthProt)

- Values: 0, 1
  0- Indicates that the user name / password method must be used by the Message Store Client (default)
  1- Indicates that the SASL methods must be used by the Message Store Client
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “AuthProt”

Node: /<x>/CPM/MessageStore/UserName

Leaf node that represents the User Identity information used by the Message Store Client to access the subscriber IMAP account on the Message Store Server

Status	Occurrence	Format	Min. Access Types
Required	One	chr	No Get, No Copy

Table 214: CPM MO sub tree addition parameters (UserName)

- Values: <username assigned to the user for access to the Message Store Server>
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “UserName”

Node: /<x>/CPM/MessageStore/UserPwd

Leaf node that represents the user password associated to his/her User Name Identity information used by the Message Store Client to access the subscriber IMAP account on the Message Store Server

Status	Occurrence	Format	Min. Access Types
Required	One	chr	No Get, No Copy

Table 215: CPM MO sub tree addition parameters (UserPwd)

- Values: <password assigned to the user for access to the Message Store Server>

- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “UserPwd”

Node: /<x>/CPM/MessageStore/SyncTimer

Leaf node that represents maximum time interval between two client-triggered synchronizations towards the Message Store Server.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

Table 216: CPM MO sub tree addition parameters (SyncTimer)

- Values: <Timer value in seconds>
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “SyncTimer”

Node: /<x>/CPM/MessageStore/SMSStore

Leaf node that describes whether the client is expected to store to the Message Store Server sent or received SMS.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get, Replace

Table 217: CPM MO sub tree addition parameters (SMSStore)

- Values: This parameter can have 3 possible values:
  0- The device shall not store any sent or received SMS to the Message Store Server
  1- The device shall store to the Message Store every sent and received SMS that cannot be correlated with the Common Message Store in the RCS dedicated user folder (RCSMessageStore)
  2- The device shall store every sent and received SMS and shall not attempt to correlate with the Common Message Store in the RCS dedicated user folder (RCSMessageStore).
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “SMSStore”

Node: /<x>/CPM/MessageStore/MMSStore

Leaf node that describes whether the client is expected to store to the Message Store Server sent or received MMS.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get, Replace

Table 218: CPM MO sub tree addition parameters (MMSStore)

- Values: This parameter can have 3 possible values:
  0- The device shall not store any sent or received MMS to the Message Store Server
  1- The device shall store to the Message Store every sent and received MMS that cannot be correlated with the Common Message Store

2- The device shall store every sent and received MMS and shall not attempt to correlate with the Common Message Store.

- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “MMSStore”

Node: /<x>/CPM/Ext

An extension node for Service Provider specific parameters. Clients that are not aware of any extensions in this subtree (e.g. because they are not Service Provider specific) should not instantiate this tree.

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

Table 219: CPM MO sub tree addition Service Provider Extension Node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-cpm:5.2:EXT*
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: “Ext”

A.2.8. Capability discovery MO sub tree

This RCS specification includes the following additions as a new configuration sub tree, the capability discovery MO sub tree. Please note this sub tree is not included in any other specifications. So no other nodes from those specifications need to be added:

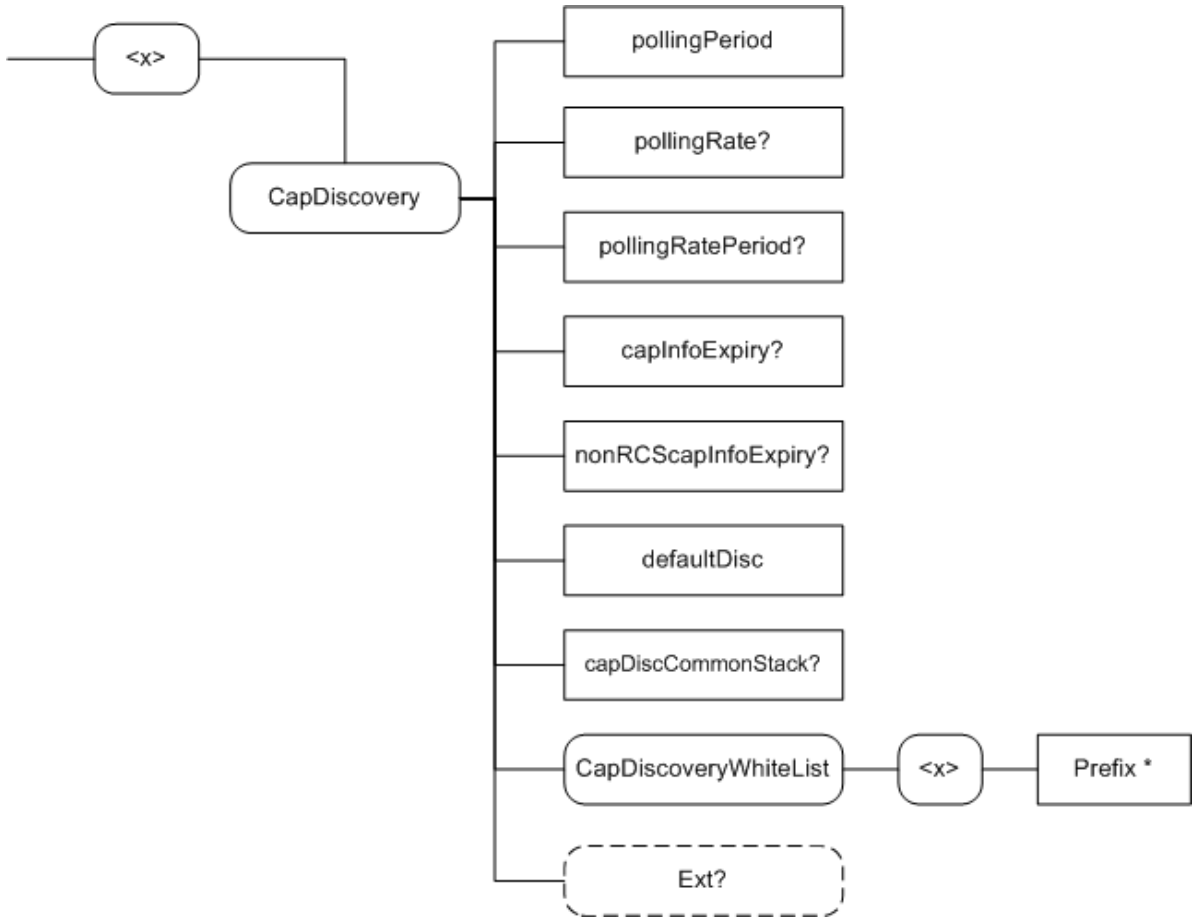


Figure 148: RCS additions, capability sub tree

The associated HTTP configuration XML structure is presented in the table below:

```
<characteristic type="CAPDISCOVERY">
        <parm name="pollingPeriod" value="X"/>
        <parm name="pollingRate" value="X"/>
        <parm name="pollingRatePeriod" value="X"/>
        <parm name="capInfoExpiry" value="X"/>
        <parm name="nonRCScapInfoExpiry" value="X"/>
        <parm name="defaultDisc" value="X"/>
        <parm name="capDiscCommonStack" value="X"/>
        <characteristic type="CapDiscoveryWhitelist">
              <characteristic type="CapDiscoveryAllowedPrefixes">
                    <parm name="Prefix1" value="X"/>
                    <parm name="Prefix2" value="X"/>
                    <parm name="Prefix3" value="X"/>
                    …
              </characteristic>
        </characteristic>
        <characteristic type="Ext"/>
</characteristic>
```

Table 220 : Capability sub tree associated HTTP configuration XML structure

Node: /<x>/CapDiscovery

Under this interior node the RCS parameters related to capability discovery are placed

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 221: Capability MO sub tree addition capability discovery node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-icapdis:5.2*
- Associated HTTP XML characteristic type: “CAPDISCOVERY”

Node: /<x>/CapDiscovery/pollingPeriod

Leaf node that describes the timer in seconds between querying all the contacts in the address book to update the capabilities.

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get, Replace

Table 222: Capability MO sub tree addition parameters (pollingPeriod)

- Values: The time in seconds. If it is set to 0, the periodic capability update (polling) is not performed
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “pollingPeriod”

Node: /<x>/CapDiscovery/pollingRatePeriod

Leaf node that indicates, in seconds, the period duration for the calculation of the authorized number of capability query requests during this period

It is only instantiated in case PollingPeriod is set to a value greater than zero.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

Table 223: Capability MO sub tree addition parameters (pollingRatePeriod)

- Values: The period in seconds.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “pollingRatePeriod”

Node: /<x>/CapDiscovery/pollingRate

Leaf node that indicates the maximum capability query operations that are authorized globally for the User Agent during each period (period parameter defined in the previous pollingRatePeriod node).

It is only instantiated in case PollingPeriod is set to a value greater than zero.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

Table 224: Capability MO sub tree addition parameters (pollingRate)

- Values: integer that represents the total amount of capability query operations per each period, independently of the number of contacts that have to be query.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “pollingRate”

Node: /<x>/CapDiscovery/capInfoExpiry

Leaf node that describes the validity of the capability information stored in the terminal in seconds

It shall be instantiated in case PollingPeriod is set to a value greater than zero.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get, Replace

Table 225: Capability MO sub tree addition parameters (capInfoExpiry)

- Values: The time in seconds.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “capInfoExpiry”

Node: /<x>/CapDiscovery/nonRCScapInfoExpiry

Leaf node that describes how long a non RCS contact shall be prevented from being queried for its capabilities.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get, Replace

Table 226: Capability MO sub tree addition parameters (nonRCScapInfoExpiry)

- Values: The time in seconds.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “nonRCScapInfoExpiry”

Node: /<x>/CapDiscovery/defaultDisc

Leaf node that describes the default capability and new user discovery mechanism used by the terminal (Presence or Options).

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get, Replace

Table 227: Capability MO sub tree addition parameters (defaultDisc)

- Values: 0, the default mechanism employed for capability discovery and new users will be OPTIONS. 1, the default mechanism employed for capability discovery and new users will be Presence
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “defaultDisc”

Node: /<x>/CapDiscovery/capDiscCommonStack

Leaf node that describes the interworking approach for the capability discovery. Please note this is only instantiated when the defaultDisc parameter is set to 1.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get, Replace

Table 228: Capability MO sub tree addition parameters (capDiscCommonStack)

- Values:
  0, the fallback to SIP OPTIONS mechanism remains disabled.
  1, the fallback to SIP OPTIONS mechanism remains enabled.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “capDiscCommonStack”

Node: /<x>/CapDiscovery/CapDiscoveryWhiteList

A Placeholder interior node for the Capability Discovery white list configuration

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 229: Capability MO sub tree addition Capability Discovery White List node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-icapdis:5.2:capdiswhitelist*
- Associated HTTP XML characteristic type: “CapDiscoveryWhiteList”

Node: /<x>/CapDiscovery/CapDiscoveryWhiteList/<x>

A Placeholder interior node where to place 1 or more Prefix leaf nodes

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 230: Capability MO sub tree addition CapDiscoveryAllowedPrefixes node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-icapdis:5.2:capdiswhitelist:prefixes*
- Associated HTTP XML characteristic type: “CapDiscoveryAllowedPrefixes”

Node: /<x>/CapDiscovery/CapDiscoveryWhiteList/<x>/Prefix

Leaf node that represent a prefix configured by the Service Provider

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	chr	No Get, No Copy

Table 231: Capability MO sub tree addition parameters (Prefix)

- Values: <a Service Provider defined prefix>.
  The prefix is is used to identify phone numbers considered for the capability discovery mechanism by matching with the prefix value starting from the left. Its length can be one or more digitsand it can start with the "+" character.
  Examples: +446, +4479, 00446, 004479, 06, 079

- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “Prefix<X>” where <X> is a positive integer value

Node: /<x>/CapDiscovery/Ext

An extension node for Service Provider specific parameters. Clients that are not aware of any extensions in this subtree (e.g. because they are not Service Provider specific) should not instantiate this tree.

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

Table 232: Capability MO sub tree addition Service Provider Extension Node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-icapdis:5.2:Ext*
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: “Ext”

A.2.9. APN Configuration MO sub tree

This RCS specification includes the following additions as a new configuration sub tree, the roaming MO sub tree. Please note this sub tree is not included in any other specifications. So no other nodes from those specifications need to be added:

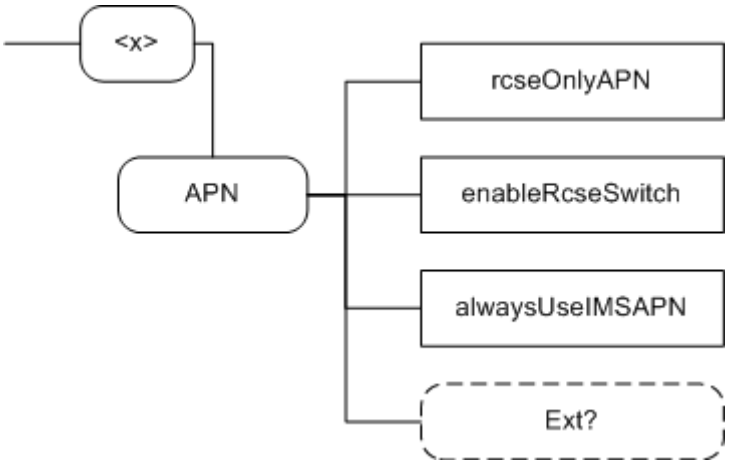


Figure 149: RCS additions, roaming sub tree

The associated HTTP configuration XML structure is presented in the table below:

```
<characteristic type=”APN”>
        <parm name=”rcseOnlyAPN” value=”X”/>
        <parm name=”enableRcseSwitch” value=”X”/>
        <parm name=”alwaysUseIMSAPN” value=”X”/>
        <characteristic type=”Ext”/>
</characteristic>
```

Table 233 : APN sub tree associated HTTP configuration XML structure

Node: /<x>/APN

Under this interior node the RCS parameters related to roaming are placed.

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 234: APN MO sub tree addition node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-apn:5.2*
- Associated HTTP XML characteristic type: “APN”

Node: /<x>/APN/rcseOnlyAPN

Leaf node that describes the APN to be used as the RCS roaming APN (as described in section 2.9.1.4)

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get, Replace

Table 235: APN MO sub tree addition parameters (rcseOnlyAPN)

- Values: The APN name or the identifier used on the phone for the RCS only APN
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “rcseOnlyAPN”

Node: /<x>/APN/enableRcseSwitch

Leaf node that describes whether or not to show the RCS enabled/disabled switch permanently as described in section 2.9.1.4

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get, Replace

Table 236: APN MO sub tree addition parameters (enableRcseSwitch)

- Values:
  1, the switch is shown permanently.
  0, the switch is only shown during roaming.
  -1, the switch is never shown.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: “enableRcseSwitch”

Node: /<x>/APN/alwaysUseIMSAPN

Leaf node that controls in what circumstances a device that is not in RCS-VoLTE or RCS-VoHSPA mode shall use the IMS APN when available.

If not instantiated, the device shall behave as if it was set to 0.

In case the device cannot support the IMS APN, it shall ignore this parameter

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get, Replace

Table 237: APN MO sub tree addition parameters (alwaysUseIMSAPN)

- Values: -1, 0, 1
  -1, The device shall never use the IMS APN for RCS. The device shall behave from

RCS perspective as a device that doesn't support the IMS APN as described in section 2.9.1.4.

NOTE: This value for the setting shall not be used for devices that are configured to use RCS-VoLTE or RCS-VoHSPA mode when available as it would result in the device having to unregister on the IMS APN and re-register on the new APN to be used.

0, The device shall use the IMS APN for RCS when the device's internet access is using cellular coverage. When the device's internet access uses non-cellular coverage, RCS shall register over that same non-cellular connection and unregister over the IMS APN, in case it was still registered.
1, The device shall always use the IMS APN for RCS whenever it is in cellular coverage, even if non-cellular access is available.

- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: "alwaysUseIMSAPN"

Node: /<x>/APN/Ext

An extension node for Service Provider specific parameters. Clients that are not aware of any extensions in this subtree (e.g. because they are not Service Provider specific) should not instantiate this tree.

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

Table 238: APN MO sub tree addition Service Provider Extension Node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-apn:5.2:Ext*
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: "Ext"

A.2.10. Other RCS Configuration MO sub tree

This RCS specification includes the following additions as a new configuration sub tree, containing the remaining RCS configuration parameters. Please note this sub tree is not included in any other specifications. So no other nodes from those specifications need to be added:

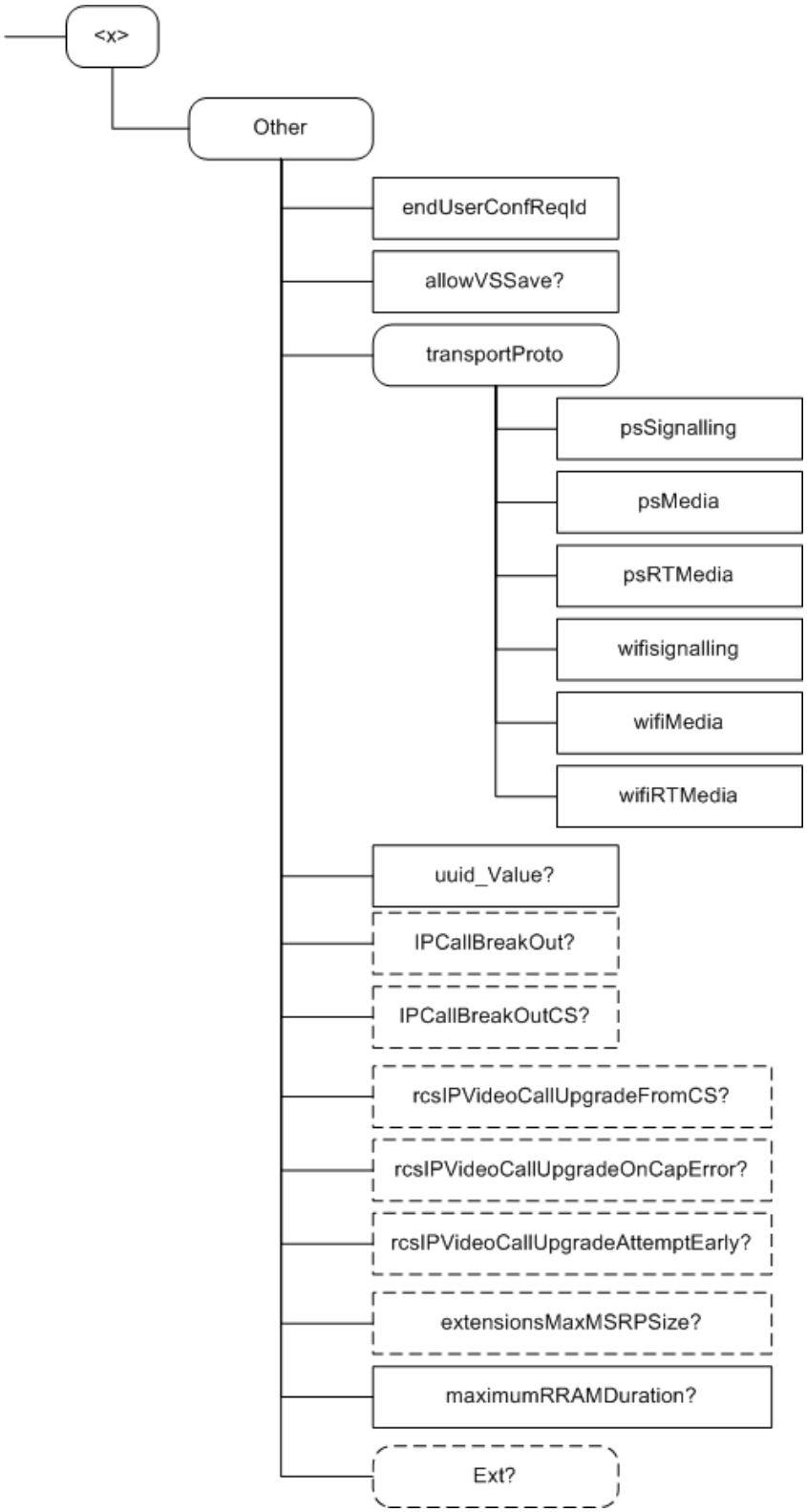


Figure 150: RCS additions, other sub tree

The associated HTTP configuration XML structure is presented in the table below:

```
<characteristic type="OTHER">
        <parm name="endUserConfReqId" value="X"/>
        <parm name="allowVSSave" value="X"/>
        <characteristic type=" transportProto">
            <parm name="psSignalling" value="X"/>
            <parm name="psMedia" value="X"/>
            <parm name="psRTMedia" value="X"/>
            <parm name="wifiSignalling" value="X"/>
            <parm name="wifiMedia" value="X"/>
            <parm name="wifiRTMedia" value="X"/>
        </characteristic>
        <parm name="uuid_Value" value="X"/>
        <parm name="IPCallBreakOut" value="X"/>
        <parm name="IPCallBreakOutCS" value="X"/>
        <parm name="rcsIPVideoCallUpgradeFromCS" value="X"/>
        <parm name="rcsIPVideoCallUpgradeOnCapError" value="X"/>
        <parm name="rcsIPVideoCallUpgradeAttemptEarly" value="X"/>
        <parm name="extensionsMaxMSRPSize" value="X"/>
        <parm name="maximumRRAMDuration" value="X"/>
        <characteristic type="Ext"/>
</characteristic>
```

Table 239 : Other sub tree associated HTTP configuration XML structure

Node: /<x>/Other

Under this interior node the RCS parameters which do not fit in the other categories are placed.

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 240: Other MO sub tree addition node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-other:5.2*
- Associated HTTP XML characteristic type: "OTHER"

Node: /<x>/Other/endUserConfReqId

Leaf node that describes the identity (*P-Asserted-Identity*) used for sending the end user confirmation request

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get, Replace

Table 241: Other MO sub tree addition parameters (endUserConfReqId)

- Values: Values: The identity (*P-Asserted-Identity*) used for sending the end user confirmation request
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: "endUserConfReqId"

Node: /<x>/Other/allowVSSave

Leaf node that determines whether or not the SDP attribute and value described in section 3.6.4.1.3 is included in the Video and Image Share invitations

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get, Replace

Table 242: Other MO sub tree addition parameters (allowVSSave)

- Values: -1, 0, 1
  -1- Inclusion of the attribute and value is up to the user's preference
  0- The attribute is never included (default if not provided)
  1- The attribute is always included
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML parameter ID: "allowVSSave"

Node: /<x>/Other/transportProto

Under this interior node the RCS parameters related to roaming are placed.

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

Table 243: Transport Protocol sub tree node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-other:5.2:transportProto*
- Associated HTTP XML characteristic type: "transportProto"

Node: /<x>/Other/transportProto/psSignalling

Leaf node that describes the transport protocol used to carry the signalling when connecting over PS cellular access.

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get, Replace

Table 244: Other MO sub tree addition parameters (psSignalling)

- Values: The possible values are:
  - SIPoUDP
  - SIPoTCP
  - SIPoTLS
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: "psSignalling"

Node: /<x>/Other/transportProto/psMedia

Leaf node that describes the transport protocol used to carry the media (e.g. Chat, File Transfer and Image Share services) when connecting over PS cellular access.

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get, Replace

Table 245: Other MO sub tree addition parameters (psMedia)

- Values: The possible values are:

  - MSRP
  - MSRPoTLS
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “psMedia”

Node: /<x>>/Other/transportProto/psRTMedia

Leaf node that describes the transport protocol used to carry the real time media (e.g. Video Share) when connecting over PS cellular access.

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get, Replace

Table 246: Other MO sub tree addition parameters (psRTMedia)

- Values: The possible values are:
  - RTP
  - SRTP
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “psRTMedia”

Node: /<x>/Other/transportProto/wifiSignalling

Leaf node that describes the transport protocol used to carry the signalling when connecting over Wi-Fi.

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get, Replace

Table 247: Other MO sub tree addition parameters (wifiSignalling)

- Values: The possible values are:
  - SIPoUDP
  - SIPoTCP
  - SIPoTLS
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “wifiSignalling”

Node: /<x>/Other/transportProto/wifiMedia

Leaf node that describes the transport protocol used to carry the media (e.g. Chat, File Transfer and Image Share services) when connecting over Wi-Fi access

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get, Replace

Table 248: Other MO sub tree addition parameters (wifiMedia)

- Values: The possible values are:

  - MSRP
  - MSRPoTLS
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “wifiMedia”

Node: /<x>/Other/transportProto/wifiRTMedia

Leaf node that describes the transport protocol used to carry the real time media (e.g. Video Share) when connecting over Wi-Fi access.

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get, Replace

Table 249: Other MO sub tree addition parameters (wifiRTMedia)

- Values: The possible values are:
  - RTP
  - SRTP
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “wifiRTMedia”

Node: /<x>/Other/uuid_Value

Leaf node that describes a UUID which is required for the sip.instance multidevice approach as described in sections 2.4.2 and 2.11. In this case the UUID is generated by the Service Provider network following the algorithm described in [RFC4122].

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get, Replace

Table 250: Other MO sub tree addition parameters (uuid_Value)

- Values: A string containing the UUID value
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “uuid_Value”

Node: /<x>/Other/IPCallBreakOut

Leaf node that tells a device in RCS-AA mode whether it can initiate IP Voice Calls (i.e. not carrying the *+g.gsma.rcs.ipcall* feature tag according to the NOTE in the row for section 2.2.4 of [PRD-IR.92] in Table 3) even if the recipient user does not show service capability for voice calls

The node is required in devices that can function in RCS-AA mode.

The node will not be instantiated if the service provider does not support the IP Voice Call service.

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get, Replace

Table 251: Other MO sub tree addition parameters (IPCallBreakOut)

- Values: 0, 1
  0- An IP Voice Call cannot be initiated if an error (480/404) is returned to a request for the recipient user's service capabilities or service capabilities do not indicate the support of IP Voice Call
  1- An IP Voice Call to be initiated even if an error (480/404) is returned to a request for the recipient user's service capabilities or service capabilities do not indicate the support of IP Voice Call
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: "IPCallBreakOut"

Node: /<x>/Other/IPCallBreakOutCS

Leaf node that tells a device in RCS-CS mode whether it can initiate IP Voice Calls (i.e. not carrying the *+g.gsma.rcs.ipcall* feature tag according to the NOTE in the row for section 2.2.4 of [PRD-IR.92] in Table 3) even if the recipient user does not show service capability for voice calls.

The node is required in devices that can function in RCS-CS mode.

The node will not be instantiated if the service provider does not support the IP Voice Call service.

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get, Replace

Table 252: Other MO sub tree addition parameters (IPCallBreakOutCS)

- Values: 0, 1
  0- An IP Voice Call cannot be initiated if an error (480/404) is returned to a request for the recipient user's service capabilities or service capabilities do not indicate the support of IP Voice Call
  1- An IP Voice Call to be initiated even if an error (480/404) is returned to a request for the recipient user's service capabilities or service capabilities do not indicate the support of IP Voice Call
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: "IPCallBreakOutCS"

Node: /<x>/Other/rcsIPVideoCallUpgradeFromCS

Leaf node that tells a device in RCS-CS mode whether it can offer to upgrade a CS call to an RCS IP Video Call

The node is required in devices that can function in RCS-CS mode.

The node will not be instantiated if the service provider does not support the RCS IP Video Call service.

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get, Replace

Table 253: Other MO sub tree addition parameters (rcsIPVideoCallUpgradeFromCS)

- Values: 0, 1
  0-does not allow upgrade of CS Voice call to RCS IP Video Call
  1-allow upgrade of CS Voice call to RCS IP Video Call

- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: “rcsIPVideoCallUpgradeFromCS”

Node: /<x>/Other/rcsIPVideoCallUpgradeOnCapError

Leaf node that tells an RCS-AA or RCS-CS device whether it can initiate an RCS IP Video Call upgrade even if service capability exchange fails with 480 Temporarily Unavailable or 408 Timeout

The node is required in devices that can function in RCS-AA mode or in RCS-CS mode.

The node will not be instantiated if the service provider does not support the RCS IP Video Call service.

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bool	Get, Replace

Table 254: Other MO sub tree addition parameters (rcsIPVideoCallUpgradeOnCapError)

- Values: 0, 1
  0- An RCS IP Video Call can only be initiated if the capability is returned from the request for the recipient user’s service capabilities
  1- An RCS IP Video Call can be initiated even if 480/408 error is returned
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: “rcsIPVideoCallUpgradeOnCapError”

Node: /<x>/Other/rcsIPVideoCallUpgradeAttemptEarly

Leaf node that tells an RCS-CS device whether it can initiate an RCS IP Video Call upgrade without first tearing down the CS voice call.

The node is required in devices that can function in RCS-CS mode.

The node will not be instantiated if the service provider does not support the RCS IP Video Call service.

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bool	Get, Replace

Table 255: Other MO sub tree addition parameters (rcsIPVideoCallUpgradeAttemptEarly)

- Values: 0, 1
  0- An RCS IP Video Call can only be initiated once the CS voice call has been torn down
  1- An RCS IP Video Call can be initiated even before the CS voice call is torn down
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: “rcsIPVideoCallUpgradeAttemptEarly”

Node: /<x>/Other/extensionsMaxMSRPSize

Leaf node that represent the maximum authorized size of an Extension to Extension message exchanged via MSRP.

It is required to be instantiated if a service provider enables Extensions by configuring a value for the extensionsIMSRouting parameter.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get

Table 256: Other MO sub tree addition parameters (extensionsMaxMSRPSize)

- Values: <content maximum size in bytes>
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: “extensionsMaxMSRPSize”

Node: /<x>/Other/maximumRRAMDuration

Leaf node that describes the maximum duration of an Audio Message.

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

Table 257: Other MO sub tree addition parameters (maximumRRAMDuration)

- Values: <Timer value in seconds>. Value equals to 0 means no limitation.
- Post-reconfiguration actions: As the client remains unregistered during configuration, there are no additional actions apart from de-registering using the old configuration and registering back (see section 2.4) using the new parameter.
- Associated HTTP XML characteristic type: “maximumRRAMDuration”

Node: /<x>/Ext

An extension node for Service Provider specific parameters. Clients that are not aware of any extensions in this subtree (e.g. because they are not Service Provider specific) should not instantiate this tree.

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

Table 258: Other MO sub tree addition Service Provider Extension Node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-other:5.2:Ext*
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration (e.g. OMA-DM/HTTP) as described in section 2.3.1.1.
- Associated HTTP XML characteristic type: “Ext”

A.2.11. Service Provider Extensions MO sub tree

This RCS specification includes the following additions as a new and optional configuration sub tree, the Service Provider extensions MO sub tree. This tree should not instantiate by clients that are not aware of any extensions in this tree. Please note this sub tree is not included in any other specifications. So no other nodes from those specifications need to be added:

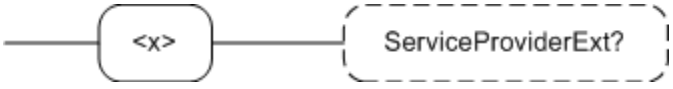


Figure 151: RCS additions, Service Provider Extensions sub tree

The associated HTTP configuration XML structure is presented in the table below:

```
<characteristic type="SERVICEPROVIDEREXT"/>
```

Table 259 : Service Provider Extensions sub tree associated HTTP configuration XML structure

Node: /<X>/ServiceProviderExt

Under this interior node the RCS parameters related to Service Provider specific extensions are placed

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

Table 260: APN MO sub tree addition node

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-sp:5.2*
- Associated HTTP XML characteristic type: “SERVICEPROVIDEREXT”

A.3. HTTP specific configuration and behaviour

A.3.1. HTTP configuration XML structure

In addition to the parameters and characteristics type correspondences presented in the previous section, it is necessary to define the following mandatory configuration XML elements[35]:

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1">
    <characteristic type="VERS">
        <parm name=”version” value=”1”/>
        <parm name=”validity” value=”1728000”/>
    </characteristic>
    <characteristic type="TOKEN">
        <parm name=”token” value=”X”/>
    </characteristic>
    <characteristic type="MSG">                    -- This section is OPTIONAL
        <parm name=”title” value=”Example”/>
        <parm name=”message” value=”Hello world”/>
        <parm name=”Accept_btn” value=”X”/>
        <parm name=”Reject_btn” value=”X”/>
    </characteristic>
    <characteristic type="APPLICATION">
        <parm name=”AppID” value=”ap2001”/>
        <parm name=”Name” value=”IMS Settings”/>
        <parm name=”AppRef” value=”IMS-Settings”/>
            …       -- see section A.2.2
    </characteristic>
    <characteristic type="APPLICATION">
        <parm name=”AppID” value=”ap2002”/>
        <parm name=”Name” value=”RCS settings”/>
        <parm name=”AppRef” value=”RCSe-Settings”/>
        <characteristic type=”IMS”>
            <parm name=”To-AppRef” value=”IMS-Settings”/>
        </characteristic>
        <characteristic type=”SERVICES”>
            …       -- See section A.2.1
        </characteristic>
        <characteristic type=”PRESENCE”>
            …       -- See section A.2.3
        </characteristic>
        <characteristic type=”XDMS”>
            …       -- See section A.2.4
        </characteristic>
        <characteristic type=”SUPL”>
            …       -- See section A.2.5
        </characteristic>
        <characteristic type=”IM”>
            …       -- See section A.2.6
        </characteristic>
        <characteristic type=”CPM”>
            …       -- See section A.2.7
        </characteristic>
```

[35] Please note the AppID’s used in the example are provided for reference only as they have not been reserved.

```
            <characteristic type="CAPDISCOVERY">
                …       -- See section A.2.8
            </characteristic>
            <characteristic type="APN">
                …       -- See section A.2.9
            </characteristic>
            <characteristic type="OTHER">
                …       -- See section A.2.10
            </characteristic>
            <characteristic type="SERVICEPROVIDEREXT">
                …       -- See section A.2.11
            </characteristic>
        </characteristic>
</wap-provisioningdoc>
```

Table 261: Complete RCS HTTP configuration XML structure

A.4. Autoconfiguration XML sample

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1">
        <characteristic type="VERS">
                <parm name=”version” value=”1”/>
                <parm name=”validity” value=”1728000”/>
        </characteristic>
        <characteristic type="TOKEN">
                <parm name=”token” value=”X”/>
                <parm name=”validity” value=”X”/>
        </characteristic>
        <characteristic type="MSG">                    -- This section is OPTIONAL
                <parm name=”title” value=”Example”/>
                <parm name=”message” value=”Hello world”/>
                <parm name=”Accept_btn” value=”X”/>
                <parm name=”Reject_btn” value=”X”/>
        </characteristic>                              -- This section is OPTIONAL
        <characteristic type="APPLICATION">
                <parm name=”AppID” value=”ap2001”/>
                <parm name=”Name” value=”IMS Settings”/>
                <parm name=”AppRef” value=”IMS-Settings”/>
                <characteristic type=”ConRefs”>
                        <parm name=”ConRef” value=”X”/>
                </characteristic>
                <parm name=”PDP_ContextOperPref” value=”X”/>
                <parm name=”Timer_T1” value=”X”/>
                <parm name=”Timer_T2” value=”X”/>
                <parm name=”Timer_T4” value=”X”/>
                <parm name=”Private_User_Identity” value=”X”/>
                <characteristic type=”Public_User_Identity_List”>
                        <parm name=”Public_User_Identity” value=”X”/>
                </characteristic>
                <parm name=”Home_network_domain_name” value=”X”/>
                <characteristic type=”Ext”>
                        <parm name=”NatUrlFmt” value=”1”/>
                        <parm name=”IntUrlFmt” value=”1”/>
                        <parm name=”Q-Value” value=”1.0”/>
                        <characteristic type=”SecondaryDevicePar”>
                                <parm name=”VoiceCall” value=”0”/>
                                <parm name=”Chat” value=”0”/>
                                <parm name=”SendSms” value=”0”/>
                                <parm name=”SendMms” value=”0”/>
                                <parm name=”FileTranfer” value=”0”/>
                                <parm name=”VideoShare” value=”0”/>
                                <parm name=”ImageShare” value=”0”/>
                                <parm name=”VideoCall” value=”0”/>
                                <parm name=”GeoLocPush” value=”0”/>
                        </characteristic>
                        <parm name=”MaxSizeImageShare” value=”0”/>
                        <parm name=”MaxTimeVideoShare” value=”0”/>
                        <characteristic type=”Ext”/>
                </characteristic>
-- Continues in the next table --
```

Table 262: Complete RCS autoconfiguration XML structure (1/5)

```
-- Follows from previous table –
            <characteristic type=”ICSI_List”>
                  <parm name=”ICSI” value=”0”/>
                  <parm name=”ICSI_Resource_Allocation_Mode” value=”X”/>
            </characteristic>
            <characteristic type=”LBO_P-CSCF_Address”>
                  <parm name=”Address” value=”X”/>
                  <parm name=”AddressType” value=”X”/>
            </characteristic>
            <parm name=”Voice_Domain_Preference_E_UTRAN” value=”X”/>
            <parm name=”SMS_Over_IP_Networks_Indication” value=”X”/>
            <parm name=”Keep_Alive_Enabled” value=”X”/>
            <parm name=”Voice_Domain_Preference_UTRAN” value=”X”/>
            <parm name=”Mobility_Management_IMS_Voice_Termination” value=”X”/>
            <parm name=”RegRetryBaseTime” value=”X”/>
            <parm name=”RegRetryMaxTime” value=”X”/>
            <characteristic type=”PhoneContext_List”>
                  <parm name=”PhoneContext” value=”X”/>
                  <parm name=”Public_User_Identity” value=”X”/>
            </characteristic>
            <characteristic type=”APPAUTH”>
                  <parm name=”AuthType” value=”X”/>
                  <parm name=”Realm” value=”X”/>
                  <parm name=”UserName” value=”X”/>
                  <parm name=”UserPwd” value=”X”/>
            </characteristic>
      </characteristic>
      <characteristic type="APPLICATION">
            <parm name=”AppID” value=”ap2002”/>
            <parm name=”Name” value=”RCS settings”/>
            <parm name=”AppRef” value=”RCSe-Settings”/>
            <characteristic type=”IMS”>
                  <parm name=”To-AppRef” value=”IMS-Settings”/>
            </characteristic>
            <characteristic type=”SERVICES”>
                  <parm name=”presencePrfl” value=”X”/>
                  <parm name=”ChatAuth” value=”X”/>
                  <parm name=”GroupChatAuth” value=”X”/>
                  <parm name=”ftAuth” value=”X”/>
                  <parm name=”standaloneMsgAuth” value=”X”/>
                  <parm name=”geolocPullAuth” value=”X”/>
                  <parm name=”geolocPushAuth” value=”X”/>
                  <parm name=”vsAuth” value=”X”/>
                  <parm name=”isAuth” value=”X”/>
                  <parm name=”rcsIPVoiceCallAuth” value=”X”/>
                  <parm name=”rcsIPVideoCallAuth” value=”X”/>
                  <parm.name=”IR94VideoAUTH” value=”X”/>
                  <parm.name=”allowRCSExtensions” value=”X”/>
                  <characteristic type=”Ext”/>
            </characteristic>
            <characteristic type=”PRESENCE”>
                  <parm name=”AvailabilityAuth” value=”X”/>
-- Continues in the next table --
```

Table 263: Complete RCS autoconfiguration XML structure (2/5)

```
-- Follows from previous table –
                    <characteristic type=”FAVLINK”>
                              <parm name=”AutMa” value=”X”/>
                              <characteristic type=”LINKS”>
                                        <parm name=” OpFavUrl1” value=”X”/>
                                        <parm name=” OpFavUrl2” value=”X”/>
                                        <parm name=” OpFavUrl3” value=”X”/>
                              </characteristic>
                              <parm name=”LabelMaxLength” value=”X”/>
                    </characteristic>
                    <parm name=”IconMaxSize” value=”X”/>
                     <parm name=”NoteMaxSize” value=”X”/>
                    <characteristic type=”VIPCONTACTS”>
                              <parm name=”NonVipPollPeriodSetting” value=”X”/>
                              <parm name=”NonVipMaxPollPerPeriod” value=”X”/>
                    </characteristic>
                    <parm name=”PublishTimer” value=”X”/>
                    <parm name=”NickNameLength” value=”X”/>
                    <characteristic type=”Location”>
                              <parm name=”TextMaxLength” value=”X”/>
                              <parm name=”LocInfoMaxValidTime” value=”X”/>
                    </characteristic>
                    <characteristic type=”Ext”/>
                    <parm name=”client-obj-datalimit” value=”X”/>
                    <parm name=”content-serveruri” value=”X”/>
                    <parm name=”source-throttlepublish” value=”X”/>
                    <parm name=”max-number-ofsubscriptions-inpresence-list” value=”X”/>
                    <parm name=”service-uritemplate” value=”X”/>
                    <parm name=”RLS-URI” value=”X”/>
          </characteristic>
          <characteristic type=”XDMS”>
                    <parm name=”RevokeTimer” value=”X”/>
                    <parm name=”enablePNBManagement” value=”X”/>
                    <parm name=”enableXDMSubscribe” value=”X”/>
                    <characteristic type=”Ext”/>
                    <parm name=”XCAPRootURI” value=”X”/>
                    <parm name=”XCAPAuthenticationUserName” value=”X”/>
                    <parm name=”XCAPAuthenticationSecret” value=”X”/>
                    <parm name=”XCAPAuthenticationType” value=”X”/>
          </characteristic>
          <characteristic type=”SUPL”>
                    <parm name=”TextMaxLength” value=”X”/>
                    <parm name=”LocInfoMaxValidTime” value=”X”/>
                    <parm name=”geolocPullOpen” value=”X”/>
                    <parm name=”geolocPullApiGwAddress” value=”X”/>
                    <parm name=”geolocPullBlockTimer” value=”X”/>
                    <characteristic type=”Ext”/>
                    <parm name=”Addr” value=”X”/>
                    <parm name=”AddrType” value=”X”/>
          </characteristic>
          <characteristic type=”IM”>
                    <parm name=”imMsgTech” value=”X”/>
                    <parm name=”imCapAlwaysON” value=”X”/>
                    <parm name=”GroupChatFullStandFwd”” value=”X ”/>
                    <parm name=”GroupChatOnlyFStandFwd” value=”X”/>
-- Continues in the next table -
```

Table 264: Complete RCS autoconfiguration XML structure (3/5)

```
-- Follows from previous table –
                    <parm name=”imWarnSF” value=”X”/>
                    <parm name=”SmsFallBackAuth” value=”X”/>
                    <parm name=”imCapNonRCS” value=”X”/>
                    <parm name=”imWarnIW” value=”X”/>
                    <characteristic type=”GroupChatNonRCSWhitelist”>
                              <parm name=”imCapNonRCSGroupChat” value=”X”/>
                              <characteristic type=”GroupChatAllowedPrefixes”>
                                        <parm name=” Prefix1” value=”X”/>
                                        <parm name=” Prefix2” value=”X”/>
                                        <parm name=” Prefix3” value=”X”/>
                                        …
                              </characteristic>
                    </characteristic>
                    <parm name=”AutAccept” value=”X”/>
                    <parm name=”imSessionStart” value=”X”/>
                    <parm name=”AutAcceptGroupChat” value=”X”/>
                    <parm name=”firstMessageInvite” value=”X”/>
                    <parm name=”TimerIdle” value=”X”/>
                    <parm name=”MaxConcurrentSession” value=”X”/>
                    <parm name=”multiMediaChat” value=”X”/>
                    <parm name=”MaxSize1to1” value=”X”/>
                    <parm name=”MaxSize1toM” value=”X”/>
                    <parm name=”ChatRevokeTimer” value=”X”/>
                    <parm name=”ftWarnSize” value=”X”/>
                    <parm name=”MaxSizeFileTr” value=”X”/>
                    <parm name=”ftThumb” value=”X”/>
                    <parm name=”ftStAndFwEnabled” value=”X”/>
                    <parm name=”ftCapAlwaysON” value=”X”/>
                    <parm name=”ftAutAccept” value=”X”/>
                    <parm name=”ftHTTPCSURI” value=”X”/>
                    <parm name=”ftHTTPCSUser” value=”X”/>
                    <parm name=”ftHTTPCSPwd” value=”X”/>
                    <parm name=”ftDefaultMech” value=”X”/>
                    <characteristic type=”Ext”/>
                    <parm name=”pres-srv-cap” value=”X”/>
                    <parm name=”deferred-msg-func-uri” value=”X”/>
                    <parm name=”max_adhoc_group_size” value=”X”/>
                    <parm name=”conf-fcty-uri” value=”X”/>
                    <parm name=”exploder-uri” value=”X”/>
          </characteristic>
          <characteristic type=”CPM”>
                    <characteristic type=”StandaloneMsg”>
                              <parm name=”MaxSizeStandalone” value=”X”/>
                    </characteristic>
                    <characteristic type=”MessageStore”>
                              <parm name=”Url” value=”X”/>
                              <parm name=”AuthProt” value=”X”/>
                              <parm name=”UserName” value=”X”/>
                              <parm name=”UserPwd” value=”X”/>
                              <parm name=”SyncTimer” value=”X”/>
                              <parm name=”SMSStore” value=”X”/>
                              <parm name=”MMSStore” value=”X”/>
                    </characteristic>
-- Continues in the next table -
```

Table 265: Complete RCS autoconfiguration XML structure (4/5)

```
-- Follows from previous table –
                    <characteristic type=”Ext”/>
                </characteristic>
                <characteristic type=”CAPDISCOVERY”>
                    <parm name=”pollingPeriod” value=”X”/>
                    <parm name=”pollingRate” value=”X”/>
                    <parm name=”pollingRatePeriod” value=”X”/>
                    <parm name=”capInfoExpiry” value=”X”/>
                    <parm name=”nonRCScapInfoExpiry” value=”X”/>
                    <parm name=”defaultDisc” value=”X”/>
                    <parm name=”capDiscCommonStack” value=”X”/>
                    <characteristic type=”CapDiscoveryWhitelist”>
                        <characteristic type=”CapDiscoveryAllowedPrefixes”>
                            <parm name=”Prefix1” value=”X”/>
                            <parm name=”Prefix2” value=”X”/>
                            <parm name=”Prefix3” value=”X”/>
                            …
                        </characteristic>
                    </characteristic>
                    <characteristic type=”Ext”/>
                </characteristic>
                <characteristic type=”APN”>
                    <parm name=”rcseOnlyAPN” value=”X”/>
                    <parm name=”enableRcseSwitch” value=”X”/>
                    <parm name=”alwaysUseIMSAPN” value=”X”/>
                    <characteristic type=”EXT”/>
                </characteristic>
                <characteristic type=”OTHER”>
                    <parm name=”endUserConfReqId” value=”X”/>
                    <parm name=”allowVSSave” value=”X”/>
                    <characteristic type=” transportProto”>
                        <parm name=”psSignalling” value=”X”/>
                        <parm name=”psMedia” value=”X”/>
                        <parm name=”psRTMedia” value=”X”/>
                        <parm name=”wifiSignalling” value=”X”/>
                        <parm name=”wifiMedia” value=”X”/>
                        <parm name=”wifiRTMedia” value=”X”/>
                    </characteristic>
                    <parm name=”uuid_Value” value=”X”/>
                    <parm name=”IPCallBreakOut” value=”X”/>
                    <parm name=”IPCallBreakOutCS” value=”X”/>
                    <parm name=”rcsIPVideoCallUpgradeFromCS” value=”X”/>
                    <parm name=”rcsIPVideoCallUpgradeOnCapError” value=”X”/>
                    <parm name=”rcsIPVideoCallUpgradeAttemptEarly” value=”X”/>
                    <parm name=”extensionsMaxMSRPSize” value=”X”/>
                    <parm name=”MaximumRRAMDuration” value=”X”/>
                    <characteristic type=”Ext”/>
                </characteristic>
                <characteristic type=”SERVICEPROVIDEREXT”/>
        </characteristic>
</wap-provisioningdoc>
```

Table 266: Complete RCS autoconfiguration XML structure (5/5)

Annex B: Additional diagrams

B.1. Chat and store and forward diagrams without Auto-Accept

B.1.1. Chat without store and forward

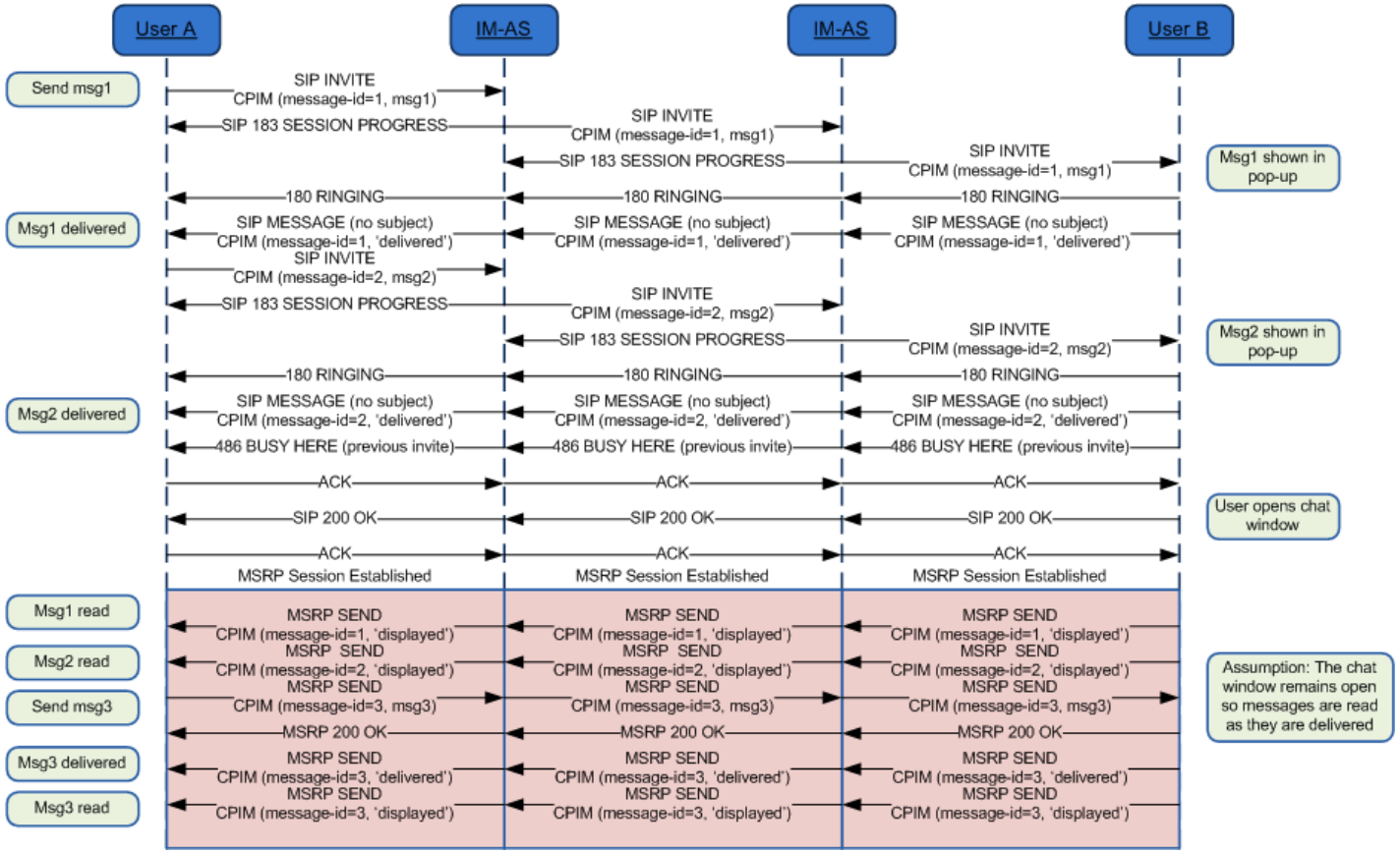


Figure 152: Chat flow without store and forward *

*: Check NOTE 1 and 15 in section B.1.19

B.1.2. Store and forward: Receiver offline

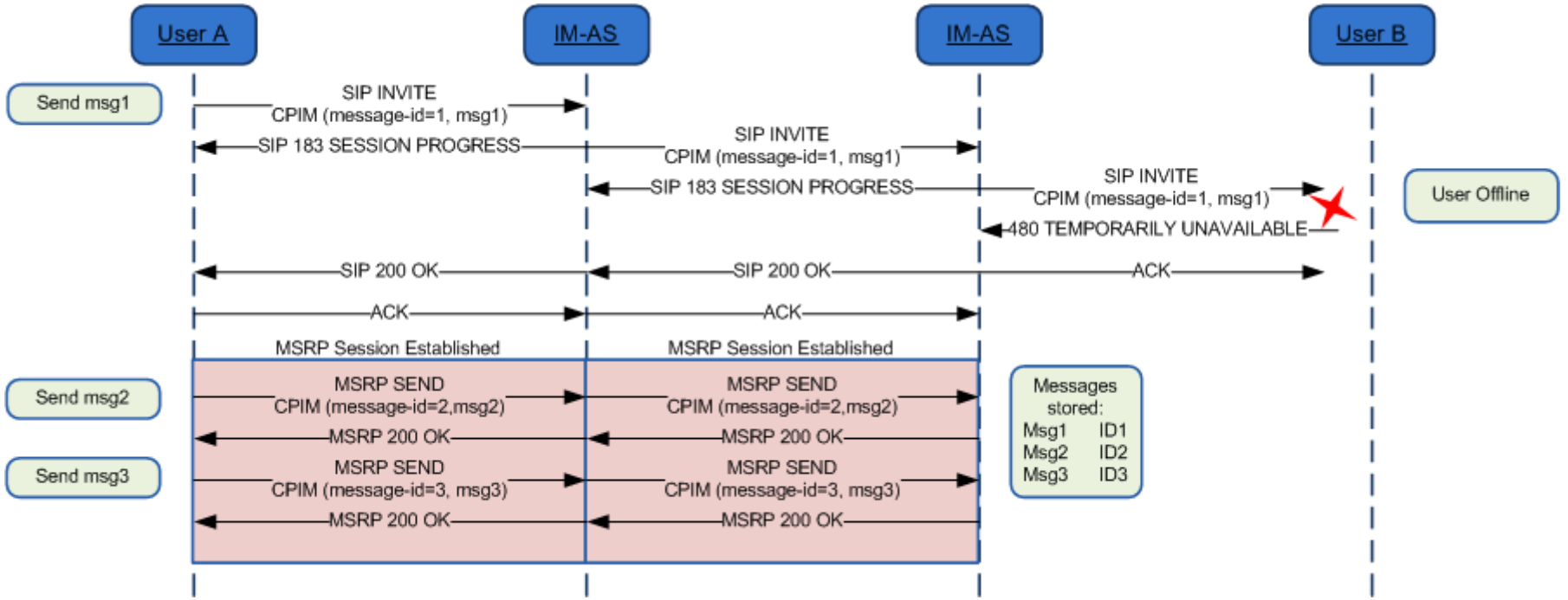


Figure 153: Store and forward: Receiver offline*

*: Check NOTE 1, 6 and 15 in section B.1.19

B.1.3. Store and forward: Message deferred delivery with sender still on an active Chat session

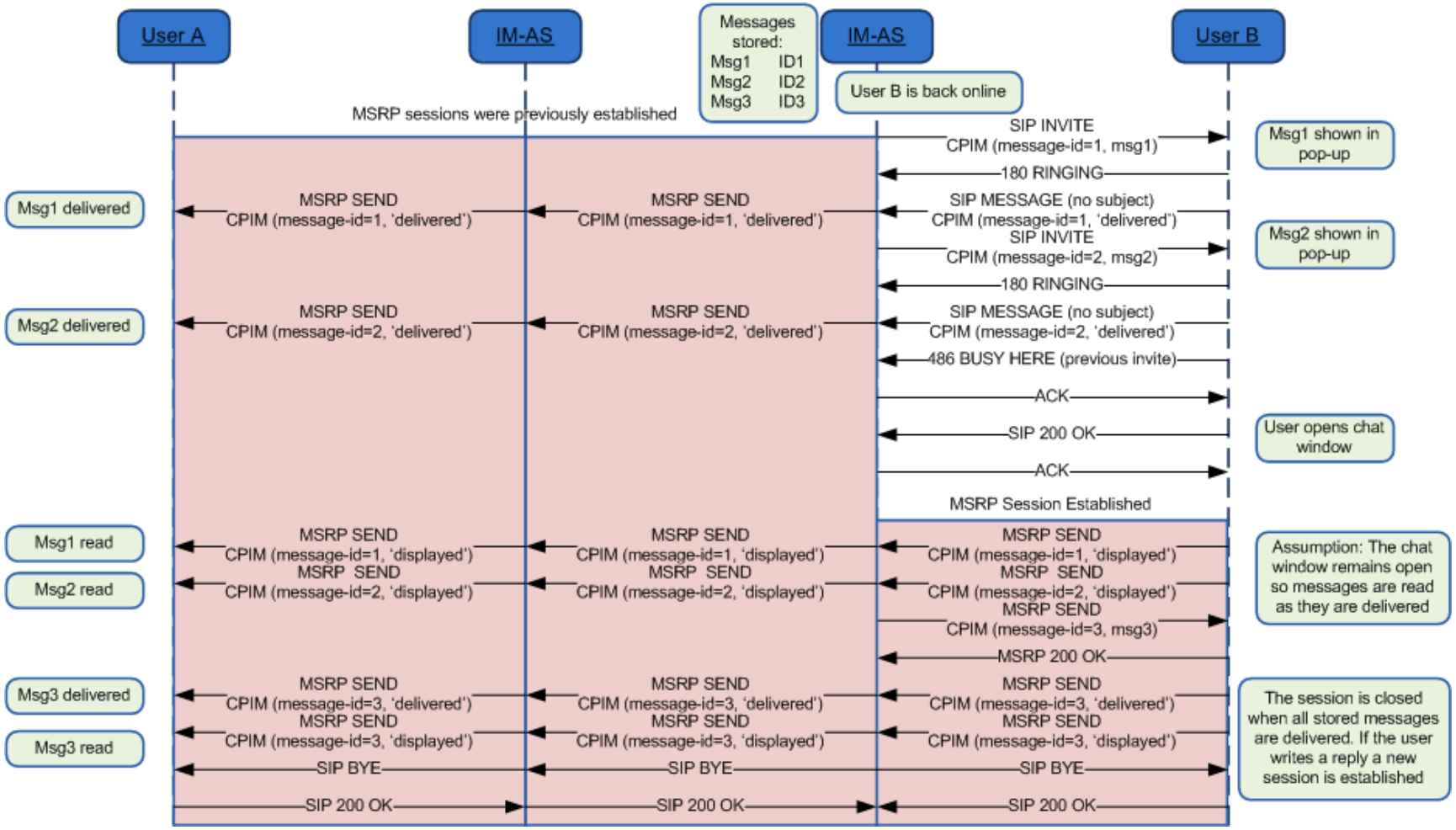


Figure 154: Store and forward: Message(s) deferred delivery with a sender still on an MSRP session*

*: Check NOTES 1, 2, 4, 7, 11 and 15 in section B.1.19

B.1.4. Store and forward: Message deferred delivery with sender online

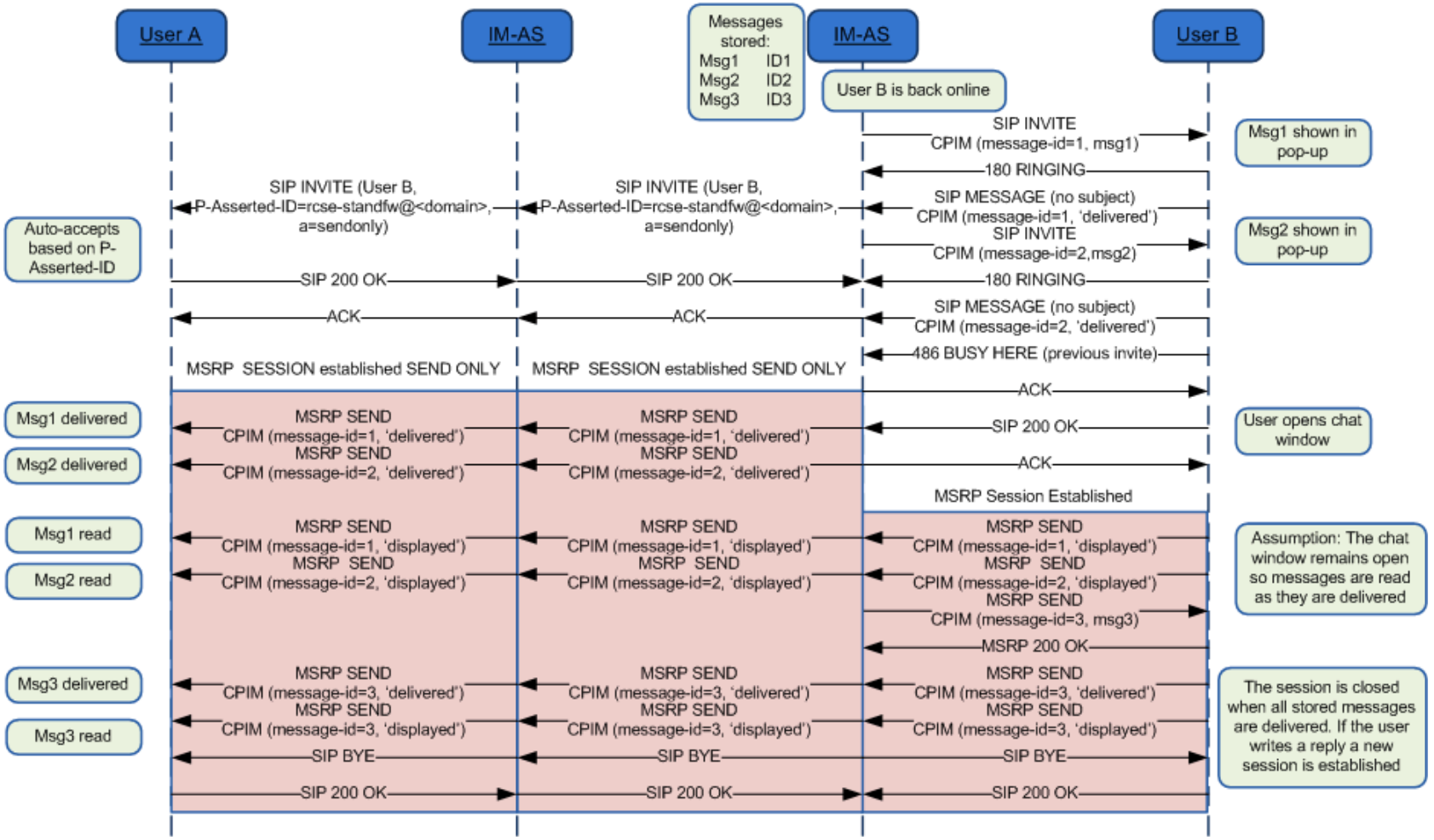


Figure 155: Store and forward: Message deferred delivery with sender online *

*: Check NOTES 1, 3, 4, 5, 7, 11, 14 and 15 in section B.1.19

B.1.5. Store and forward: Message deferred delivery with sender offline (delivery notifications)

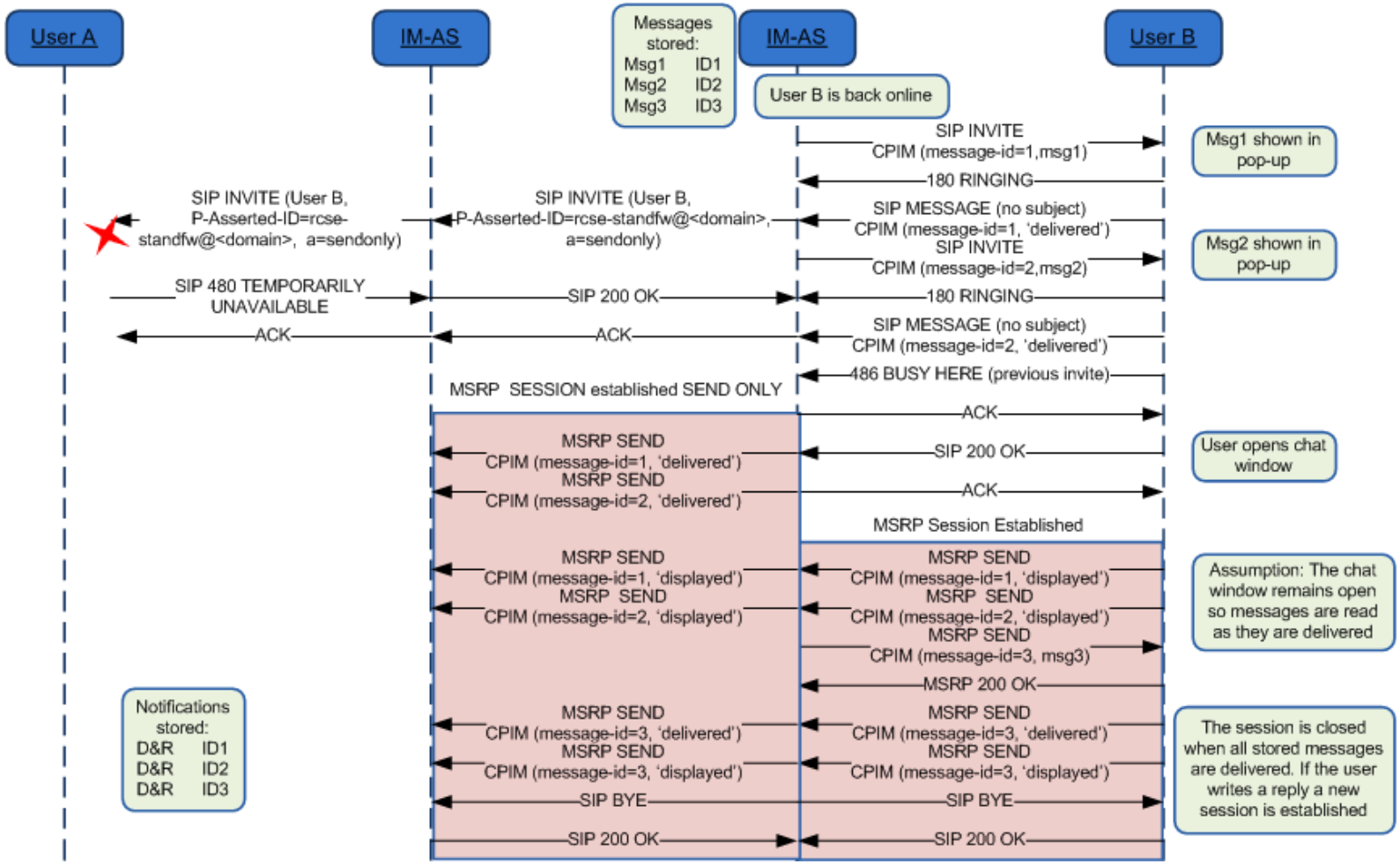


Figure 156: Store and forward: Message(s) deferred delivery with a sender offline (delivery notifications)*

*: Check NOTE 1, 5, 7, 11, 14 and 15 in section B.1.19

B.1.6. Store and forward: Notifications deferred delivery

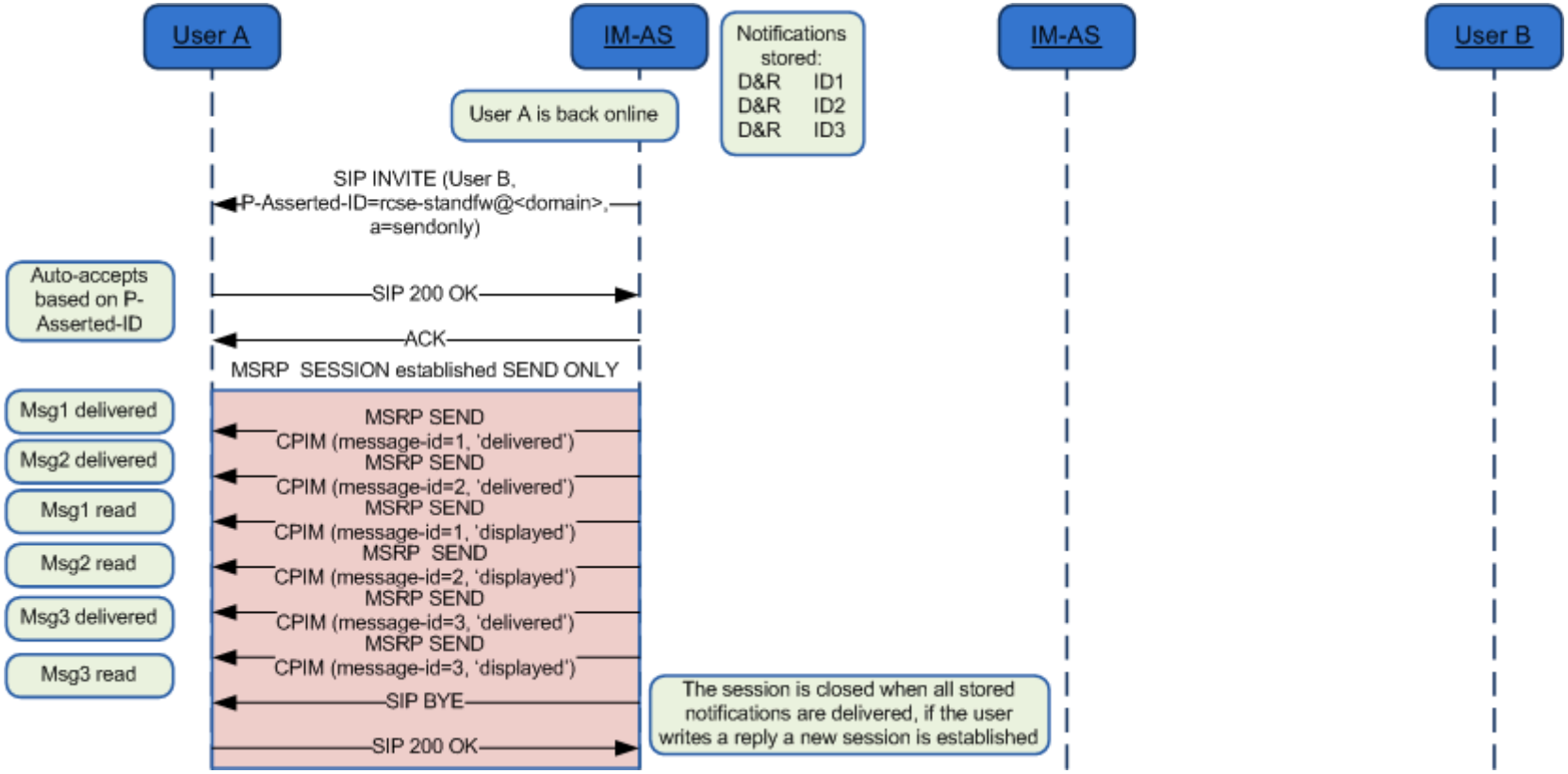


Figure 157: Store and forward: Notification(s) deferred delivery*

*: Check NOTES 1, 4, 5, 11, 14 and 15 in section B.1.19

B.1.7. Delivery of displayed notifications in an unanswered chat (without store and forward)

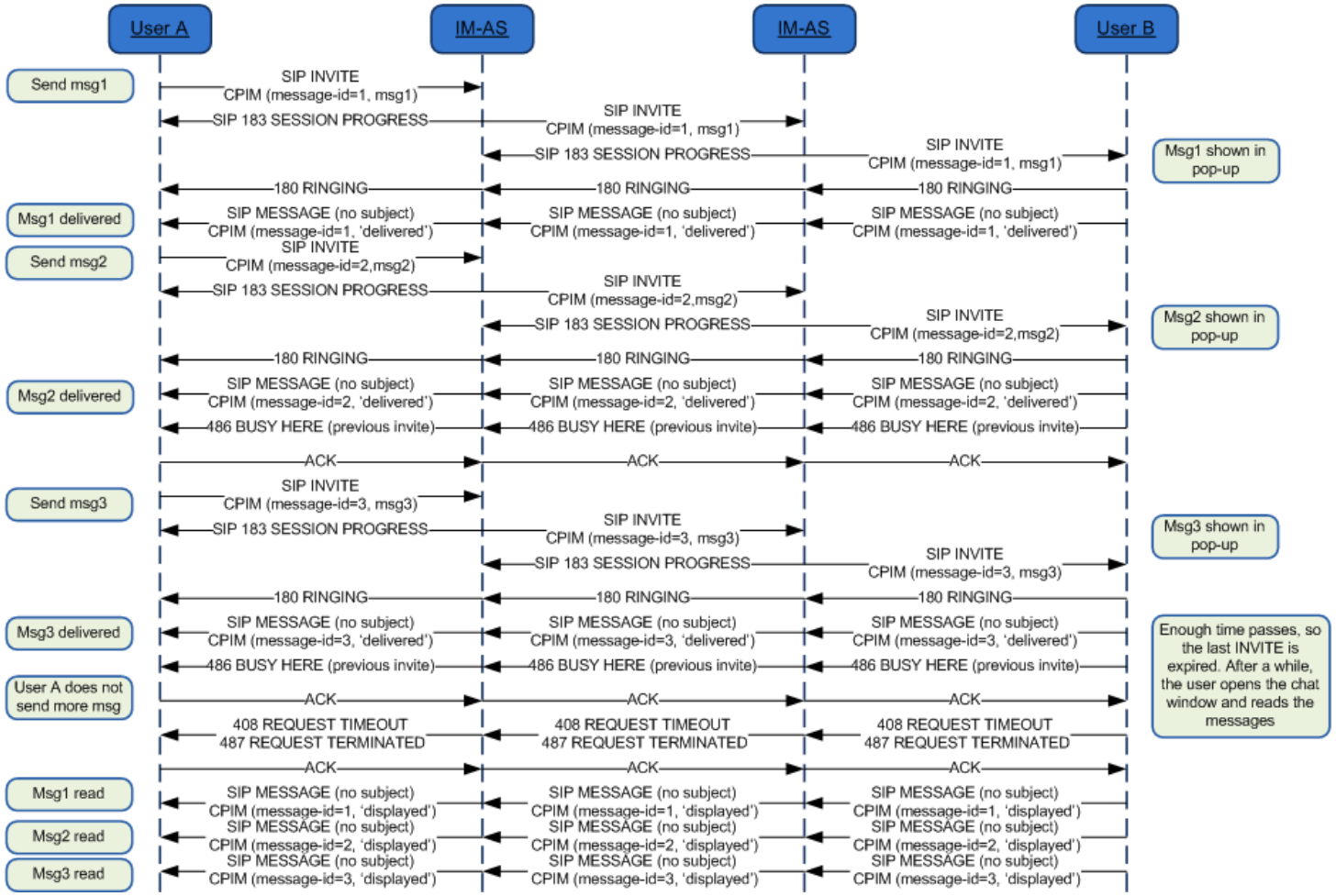


Figure 158: Delivery of displayed notifications in an unanswered chat (without store and forward)*

*: Check NOTE 1, 10 and 15 in section B.1.19

B.1.8. Store and forward: Handling errors in the receiver's side

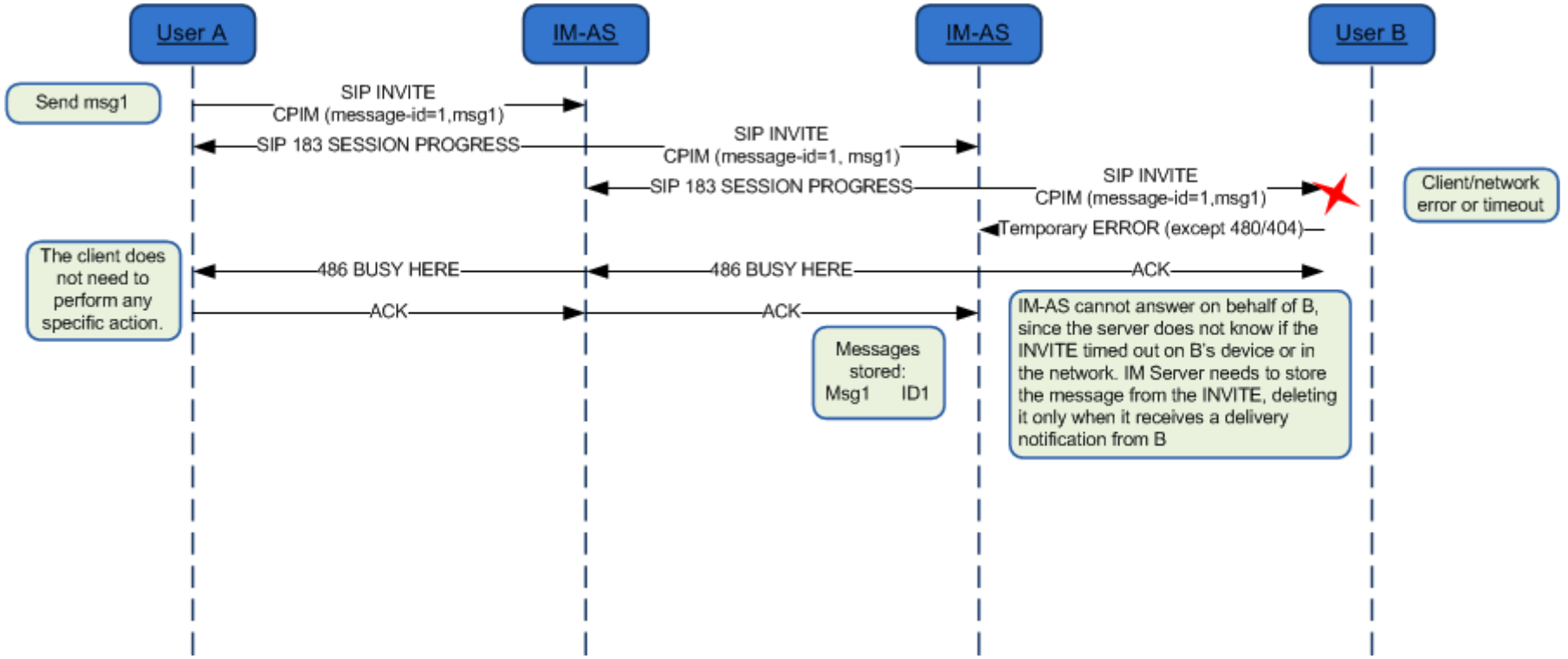


Figure 159: Store and forward: Handling errors in the receiver's side*

*: Check NOTE 15 in section B.1.19

NOTE: The error messages that are mapped to 486 Busy Here are listed in Table 55.

Also on the path between the IM-ASs (Instant Messaging Application Server i.e. the Messaging Server) similar errors could occur. In that case if the originating Messaging Server supports Store and Forward, it will behave in the same way and store the message.

B.1.9. Race conditions: Simultaneous INVITEs

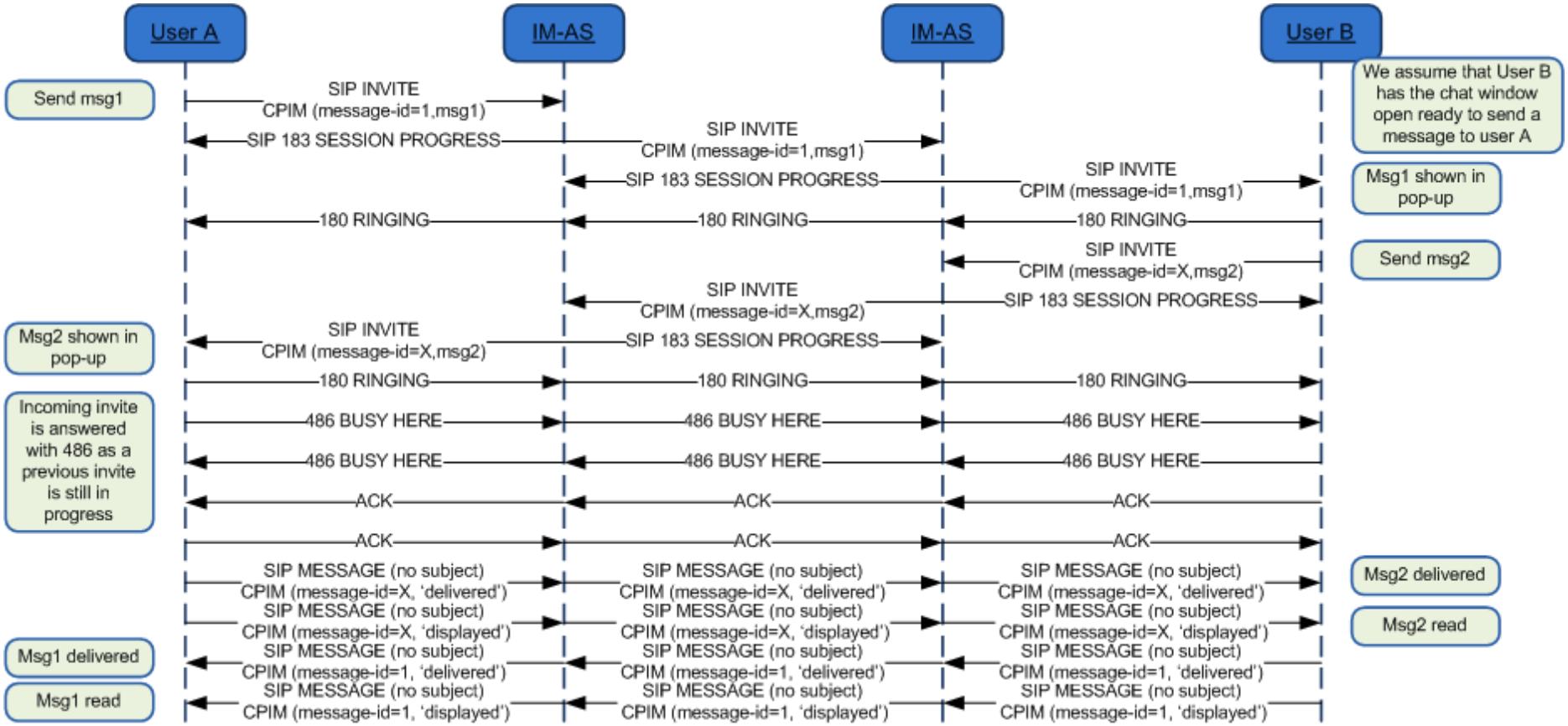


Figure 160: Store and forward race conditions: Simultaneous INVITEs*

*: Check NOTE 1 and 15 in section B.1.19

B.1.10. Race conditions: New INVITE after a session is accepted

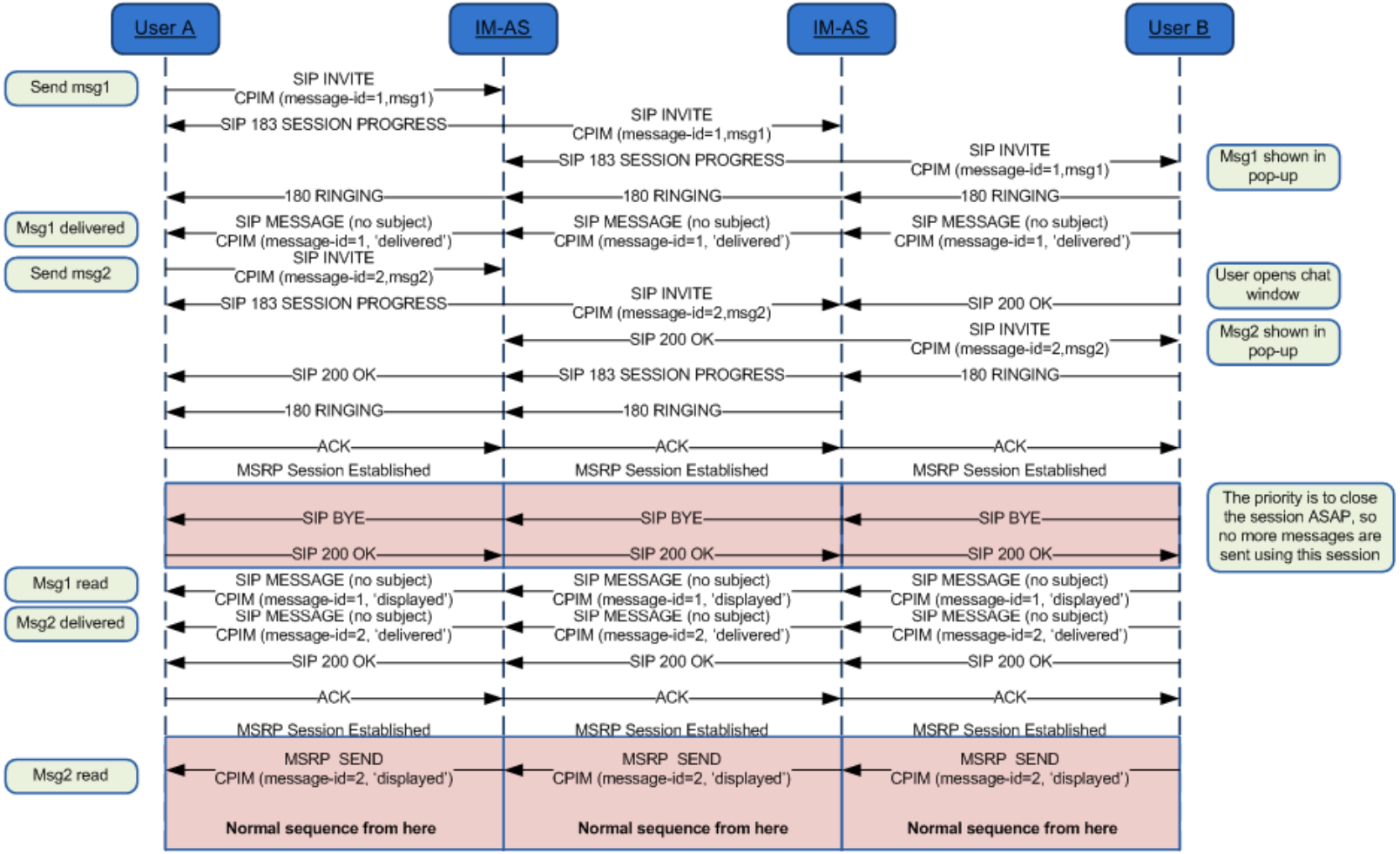


Figure 161: Store and forward race conditions: New INVITE after a session is accepted*

*: Check NOTE 1 and 15 in section B.1.19

B.1.11. Store and forward: Message(s) displayed notifications via SIP MESSAGE with sender offline

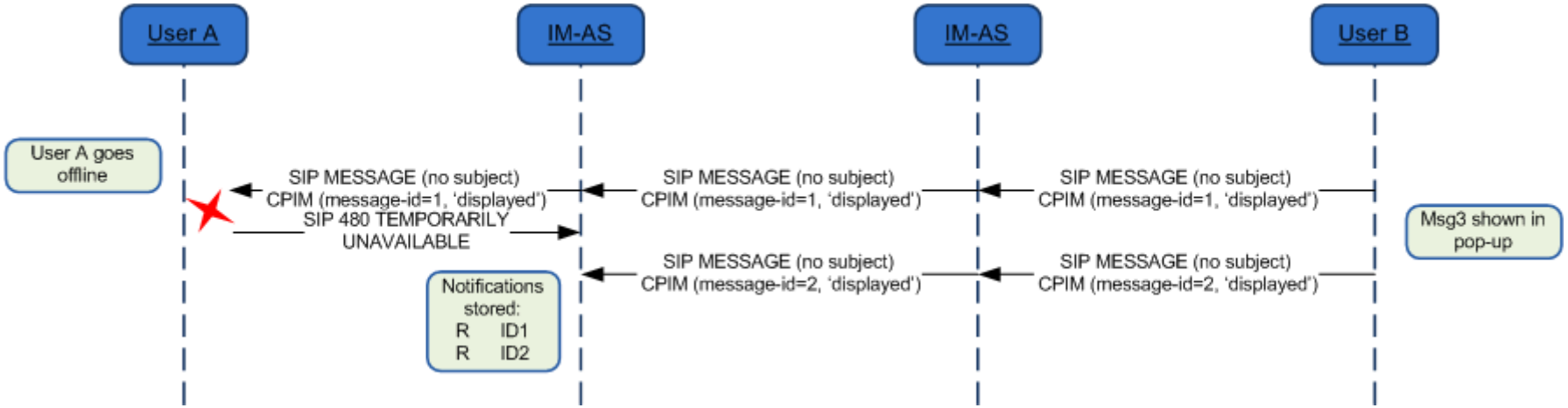


Figure 162: Store and forward: Message(s) displayed notifications via SIP MESSAGE with sender offline*

*: Check NOTES 1, 8, 9, 10 and 15 in section B.1.19

B.1.12. Interworking to SMS/MMS with automatic accept at the IWF

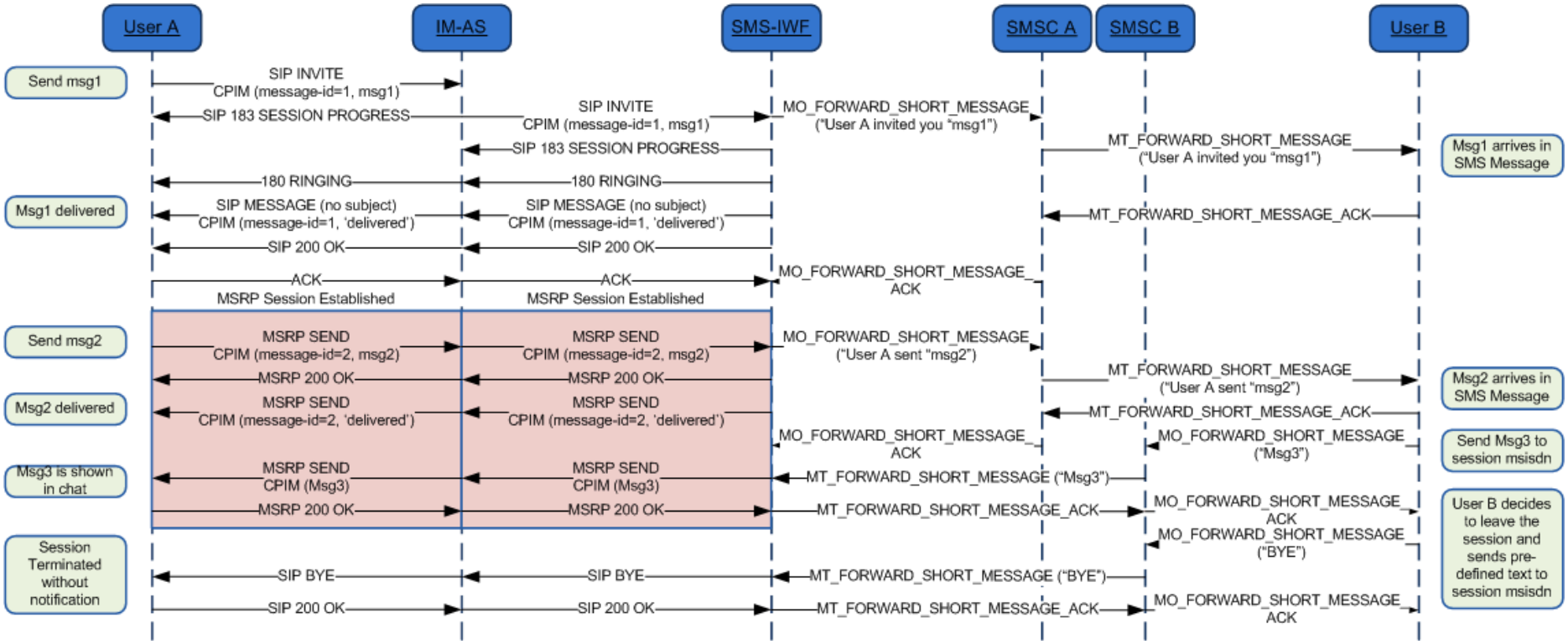


Figure 163: Interworking: Automatic acceptance on behalf of the SMS/MMS user*

*: Check NOTES 1, 12, 15 and 16 in section B.1.19

B.1.13. Interworking to SMS/MMS with manual accept

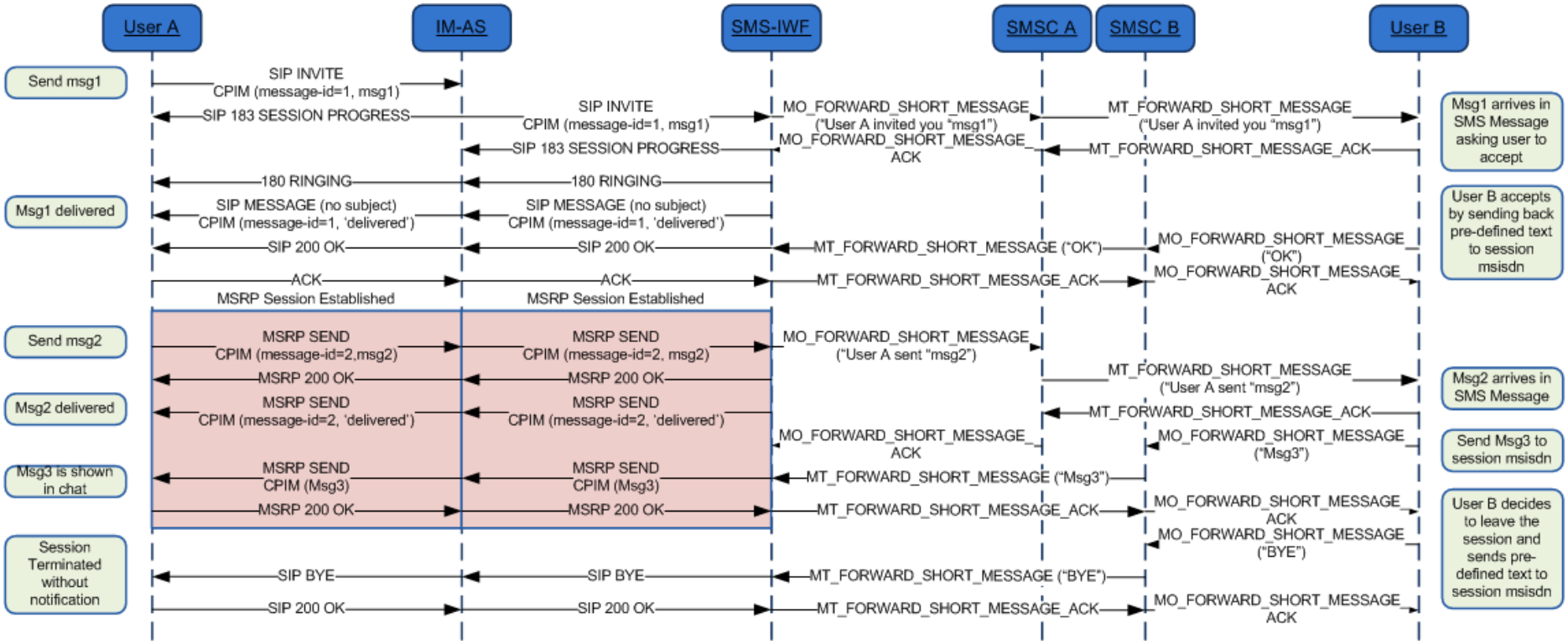


Figure 164: Interworking: manual acceptance by the SMS/MMS user*

*: Check NOTES 1, 12, 13, 15 and 16 in section B.1.19

B.1.14. Message Revoke: Successful Request

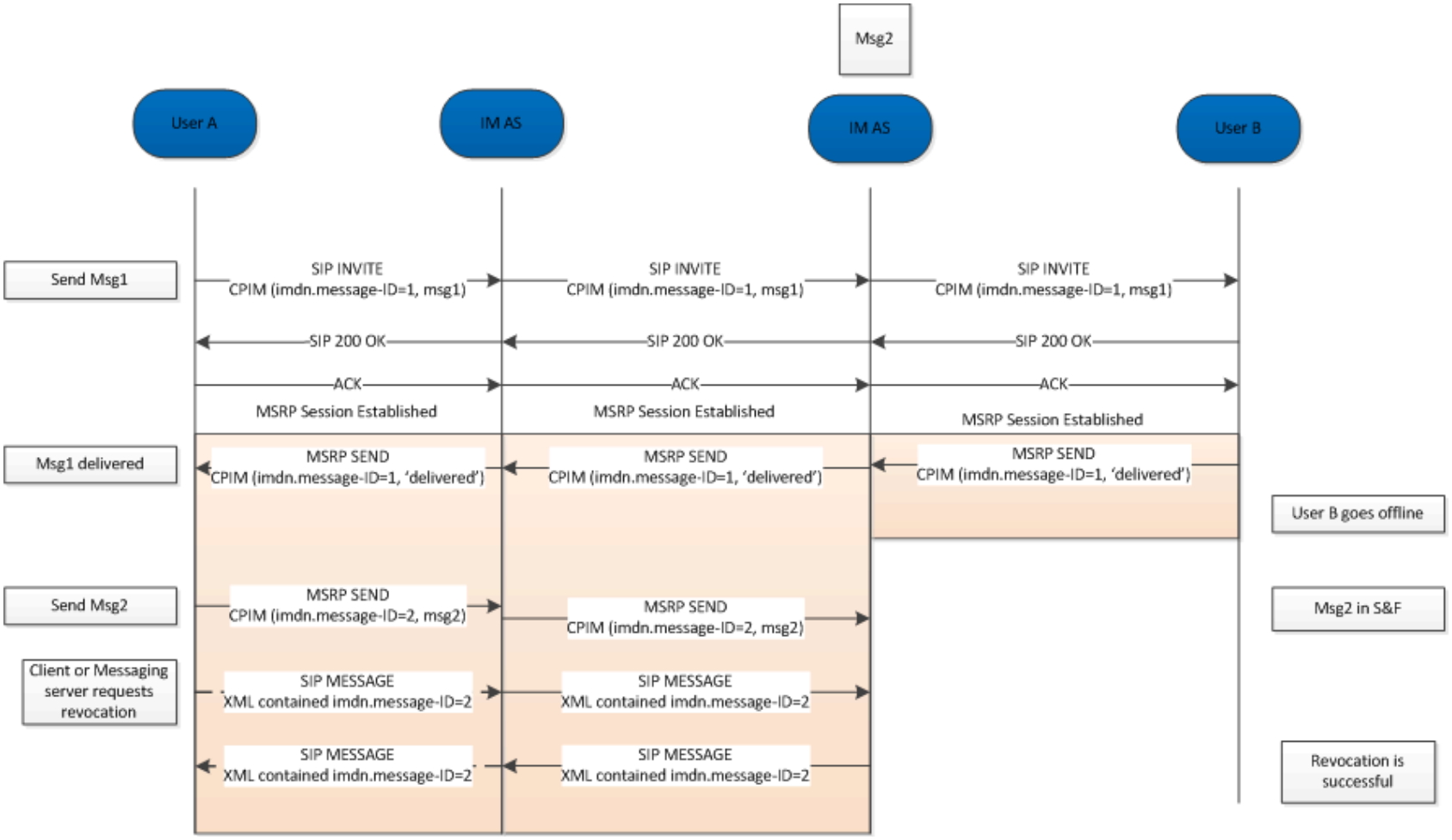


Figure 165: Message Revoke, Successful request*

*: Check NOTES 1, 6, 7 and 15 in section B.1.19

B.1.15. Message Revoke: Failed Request

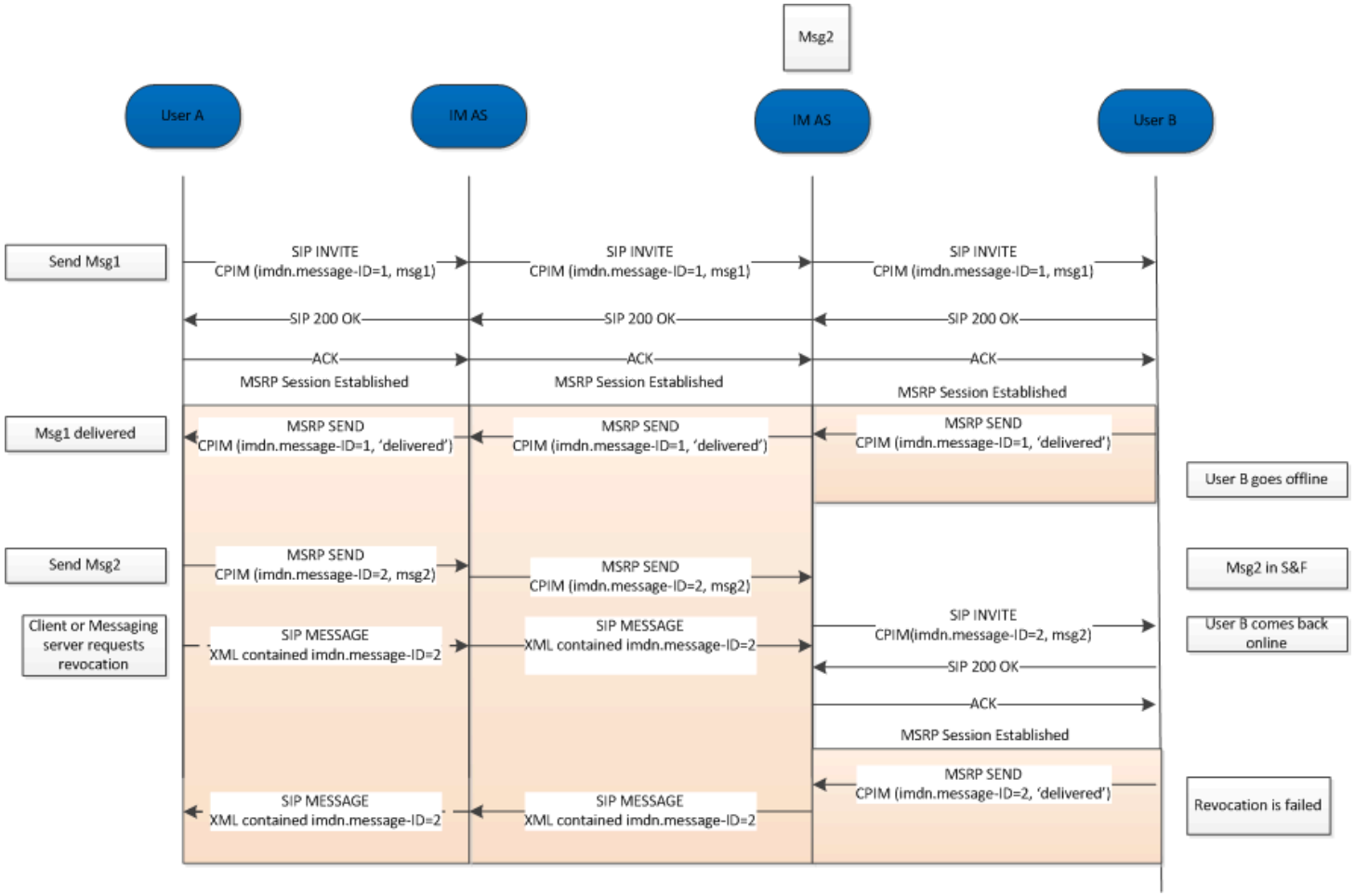


Figure 166: Message Revoke, Failed request*

*: Check NOTES 1, 6, 7 and 15 in section B.1.19

B.1.16. Rejoining a Group Chat that timed out due to inactivity

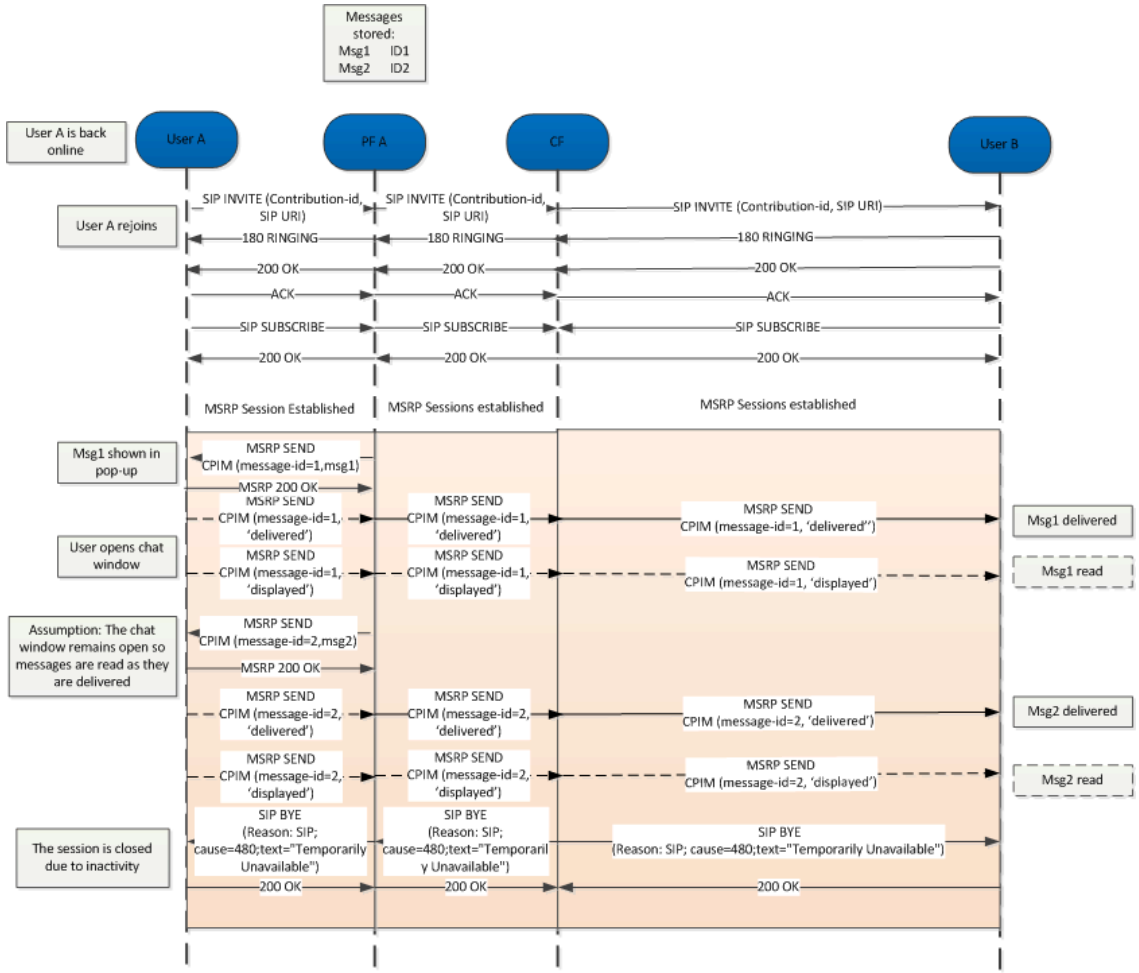


Figure 167: Rejoining a Group Chat that timed out due to inactivity *

*: Check NOTES 1, 15 and 17 in section B.1.19

B.1.17. Deliver Group Chat Messages while Chat is idle

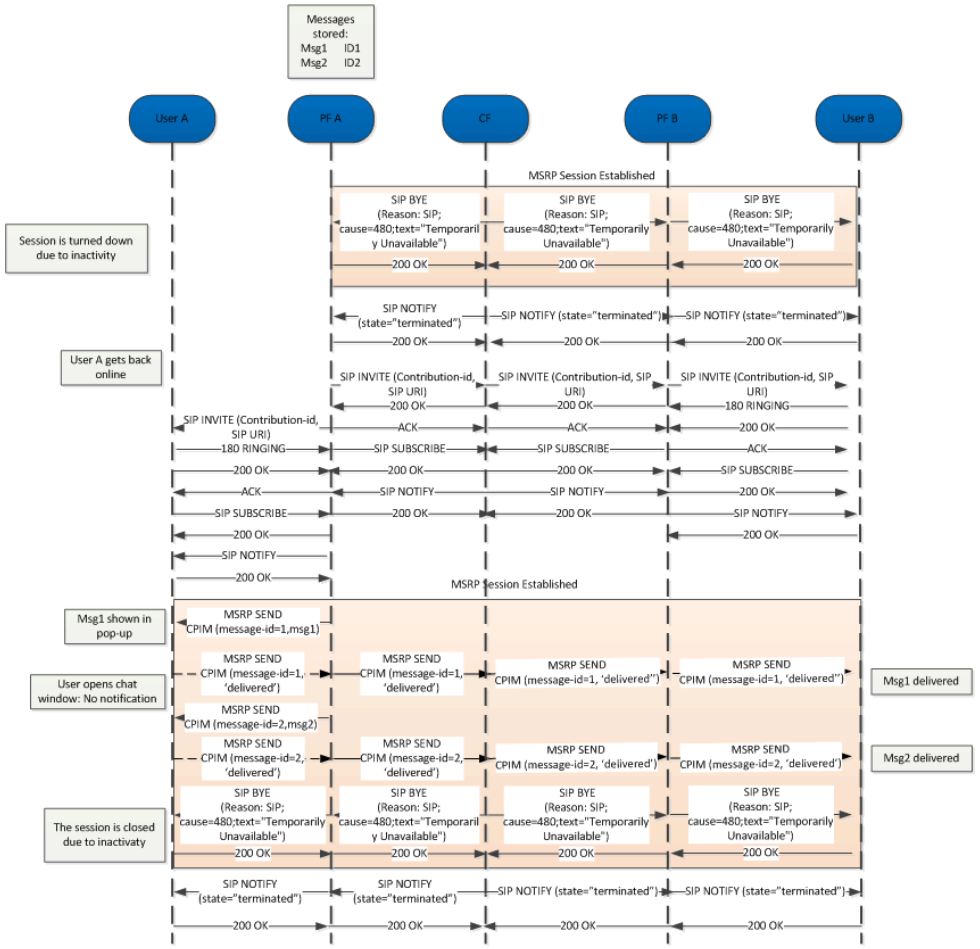


Figure 168: Deliver Group Chat Messages while Chat is idle*

*: Check NOTES 1, 15, 17, 18 and 19 in section B.1.19

B.1.18. Race Condition: user rejoins active Group Chat which is torn down due to inactivity

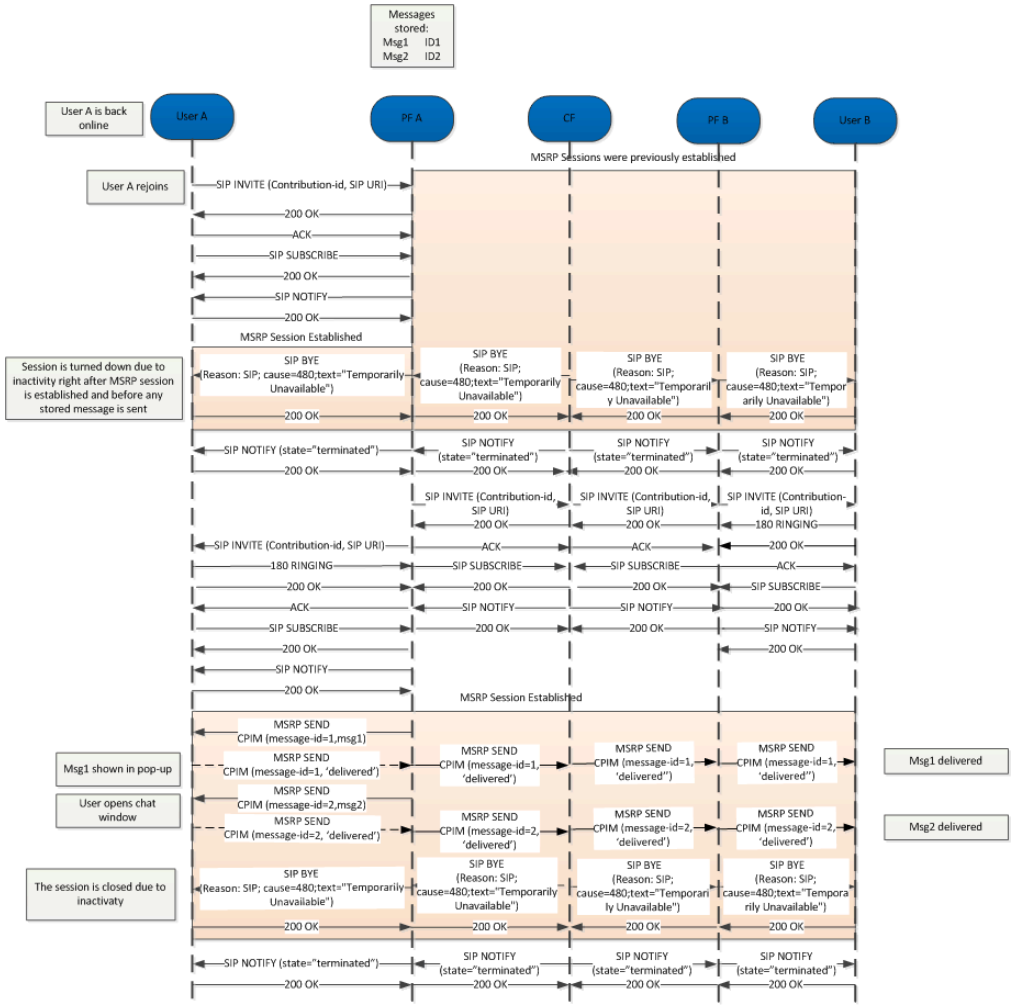


Figure 169: Rejoin in active Chat which is torn down due to inactivity*

*: Check NOTES 1, 15, 17, 18 and 19 in section B.1.19

B.1.19. Chat and store and forward diagrams: Notes

Please note the following notes apply to diagrams in section B.1:

- NOTE 1 (B.1.1, B.1.2, B.1.3, B.1.4, B.1.5, B.1.6, B.1.7, B.1.9, B.1.10, B.1.11, B.1.12, B.1.13, B.1.14, B.1.15, B.1.16, B.1.17 and B.1.18): 200 OK responses to SIP MESSAGE and MSRP SEND messages are omitted for clarity.
- NOTE 2 (B.1.3): In a multidevice scenario, if the device public GRUU in a delivery notification received from User B is different from the value for User A's device used in the ongoing MSRP session, a new session with automatic acceptance needs to be set up as specified in section 3.3.4.1.5.
- NOTE 3 (B.1.4): In a multidevice scenario, if the device public GRUU in a delivery notification received after the first INVITE is sent to User A is different from the value in the first one, a new SIP INVITE with the new device public GRUU needs to be sent towards A.
- NOTE 4 (B.1.3, B.1.4 and B.1.6): B could have to handle two incoming INVITEs, one from the Messaging Server on behalf of A to deliver messages and notifications that were stored to be forwarded, and a second one directly from A who happens to want to chat with B at the same time. B should recognize the INVITE from the Messaging Server on behalf of A and not tear it down when the new INVITE directly from A arrives: The INVITE from the Messaging Server has a Referred-By header and no isfocus tag, and the INVITE directly from A does not have a Referred-By header. Please note that the same applies to the case in which the order in which the INVITEs arrive is reversed.
- NOTE 5 (B.1.3, B.1.4, B.1.5 and B.1.6): The session established by the Messaging Server to deliver deferred messages to the destination only allows the receiver (client/device) to send back notifications (that is an INVITE with referred-by header will only allow message/imdn+xml in the CPIM part). If the user replies with a new message, then a separate session shall be established (That is if User B (the receiver) wants to reply, a new INVITE should be used) after all the deferred messages have been delivered.
- NOTE 6 (B.1.2, B.1.14 and B.1.15): In the diagram we have represented one of the possible mechanisms to detect that the user is not online (wait for the 480 response), however, there are alternative mechanisms (triggers, 3rd party registration) that can be also used by the Messaging Server for the purpose.
- NOTE 7 (B.1.3, B.1.4, B.1.5, B.1.14 and B.1.15): Note that in the scenario where the MSRP socket is closed between the Messaging Server and the Terminating client (B) in a deferred message delivery (due for instance to a small connectivity loss with the PDP context remaining active) and no re-registration takes place, if there are notifications pending (delivery or displayed) and all the deferred messages have been sent to B already (no need to open a new MSRP session), SIP MESSAGE can be used to confirm the pending delivery/display notifications that could not be sent over MSRP.
- NOTE 8 (B.1.11): Note that the deferred delivery of the display notifications stored to be forwarded in the Messaging Server will be performed as shown in section B.1.6.

- NOTE 9 (B.1.11): In the absence of a Messaging Server (neither in the sender's nor in the receiver's domain) and in the case the display notification fail to be delivered because the sender is offline, these notifications will be discarded and the receiver's client does not need to retry sending them. In any case, the next time User A manages to establish a chat session with User B, all the previous messages pending to receive the displayed notification will be marked as displayed/read.
- NOTE 10 (B.1.7 and B.1.11): In those scenarios where a Messaging Server is not available, neither in the sender's nor in the receiver's network, there is a chance that display notifications carried via SIP MESSAGE may be lost if the original sending client is offline when the receiver sends those display notifications (that is the last three messages in the diagram). To overcome this limitation, a terminal or client implementation should mark all the previous messages as displayed when a new chat message is received from the receiving user.
- NOTE 11 (B.1.3, B.1.4, B.1.5 and B.1.6): The session established by the Messaging Server to deliver deferred messages or notifications should be terminated once the all the messages and notifications have been delivered. In more detail:
    - When delivering deferred messages, the session should be terminated (by sending a BYE) either (whatever is shorter) when the display notification corresponding to the last deferred message has been received by the Messaging Server or, after a timer started on the reception of the delivered notification for the last message expires. This timer is defined by the Service Provider.
- NOTE 12 (B.1.12 and B.1.13): The predefined text for accepting and leaving a session is included for illustration purposes only as it is up to the Service Provider providing the interworking to configure an appropriate an appropriate text and announce that to the SMS/MMS user when appropriate.
- NOTE 13 (B.1.13): If the SMS (or MMS) user does not respond in time, the INVITE will have timed out and the used MSISDN may even be assigned to another session. For that reason the Messaging Server should check whether the SMS (or MMS) message comes from a user that is invited to the related session and if that is not the case or the MSISDN is not assigned to any session, a message is sent back informing the user that he cannot join the session any longer.
- NOTE 14 (B.1.4, B.1.5 and B.1.6): Whether a Messaging Server sets up a session for the delivery of notifications or sends them using SIP MESSAGE requests is up to its local policy. This could depend on factors such as the number of notifications that were stored or the number of messages for which notifications can be expected (during delivery of stored messages for instance).
- NOTE 15 (B.1.1, B.1.2, B.1.3, B.1.4, B.1.5, B.1.6, B.1.7, B.1.8, B.1.9, B.1.10, B.1.11, B.1.12, B.1.13, B.1.14, B.1.15, B.1.16, B.1.17 and B.1.18): As per [RFC5438], the message-id is conveyed in the messages via the imdn.Message-ID header and in the notifications via the value of the <message-id> element in the body of the IMDN.
- NOTE 16 (B.1.12 and B.1.13): The flows show interworking with SMS, but the flows in the SIP/MSRP part of the figure also apply when interworking with MMS.
- NOTE 17 (B.1.16, B.1.17 and B.1.18): As per sections 3.4.4.1.7 and 3.4.4.3.4.

- NOTE 18 (B.1.17 and B.1.18): The flow shows the Participating Function restarting the session before attempting the delivery. This is an implementation option to ensure that a session is established when the user sends content. The Participating Function may also choose to establish this session in parallel or only when there is actual content to be sent in the Chat.
- NOTE 19 (B.1.17 and B.1.18): The flow assumes that no display notifications were requested.

B.2. Chat and store and forward diagrams with Automatic Acceptance

B.2.1. Chat without store and forward

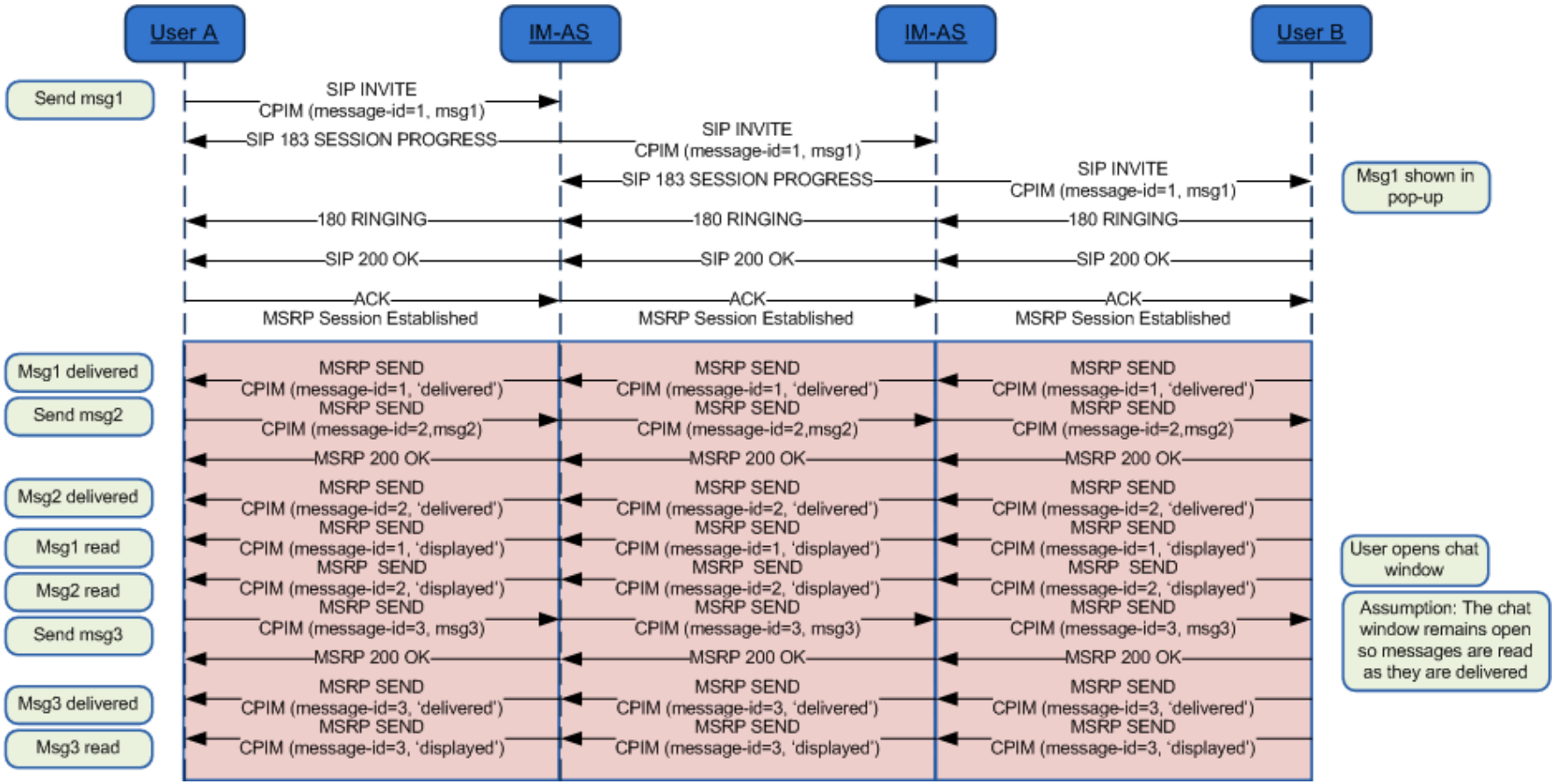


Figure 170: Chat flow without store and forward *

*: Check NOTES 1, 2, 16 and 17 in section B.2.19

B.2.2. Store and forward: Receiver offline

This case is identical to the one without automatic acceptance (see section B.1.2). NOTES 1, 2, 7 and 17 in section B.2.19 apply as well.

B.2.3. Store and forward: Message deferred delivery with sender still on an active Chat session

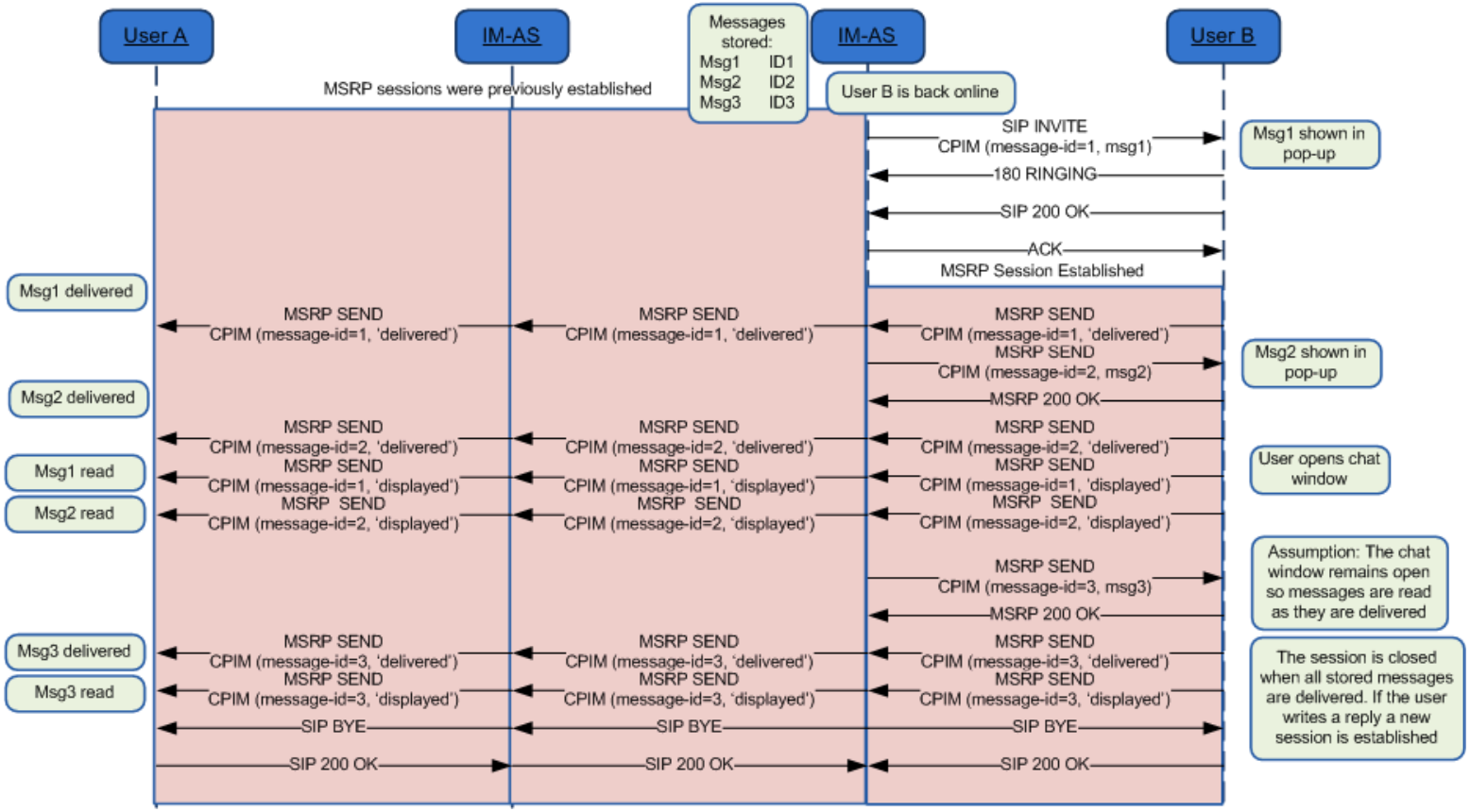


Figure 171: Store and forward: Message(s) deferred delivery with a sender still on an MSRP session*

*: Check NOTES 1, 2, 3, 5, 6, 8, 12, 16 and 17 in section B.2.19

B.2.4. Store and forward: Message deferred delivery with sender online

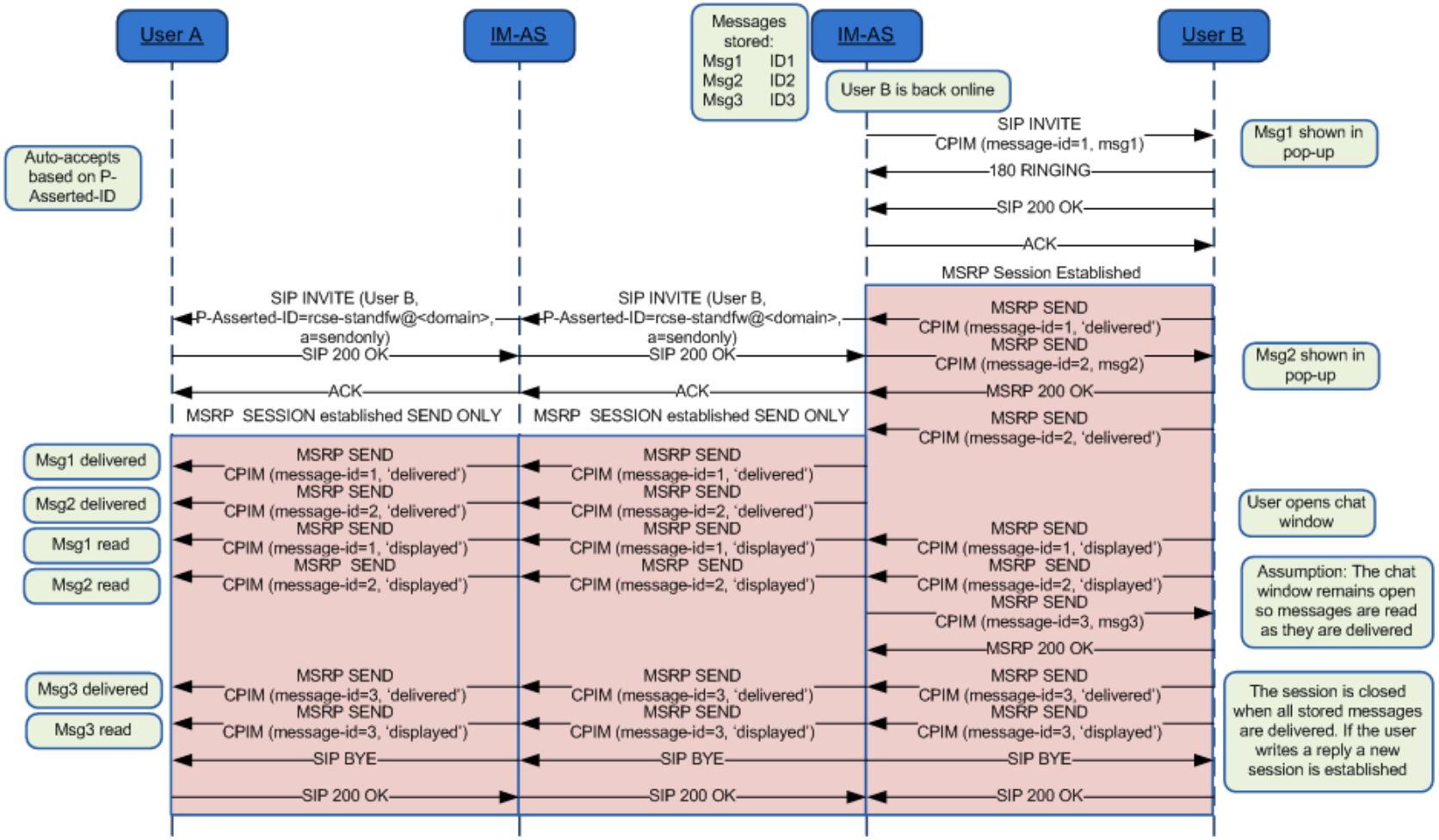


Figure 172: Store and forward: Message deferred delivery with sender online *

*: Check NOTES 1, 2, 4, 5, 6, 8, 12, 15, 16 and 17 in section B.2.19

B.2.5. Store and forward: Message deferred delivery with sender offline (delivery notifications)

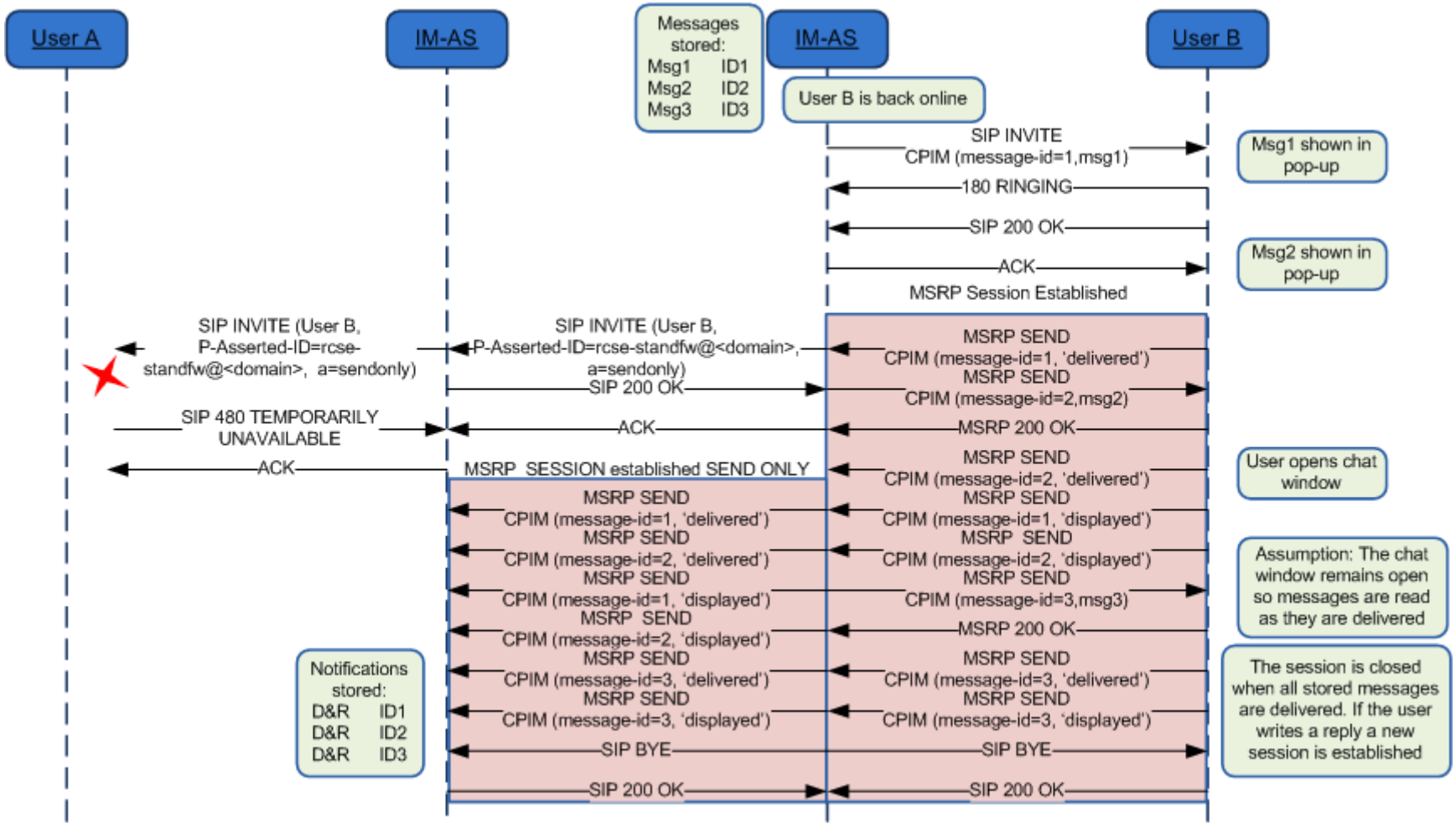


Figure 173: Store and forward: Message(s) deferred delivery with a sender offline (delivery notifications)*

*: Check NOTE 1, 2, 6, 8, 12, 15, 16 and 17 in section B.2.19

B.2.6. Store and forward: Notifications deferred delivery

This case is identical to the one without automatic acceptance (see section B.1.6). NOTES 2, 5, 6, 12, 15 and 17 in section B.2.19 apply as well.

B.2.7. Delivery of displayed notifications in an unanswered chat (without store and forward)

This case is not applicable in case of automatic acceptance.

B.2.8. Store and forward: Handling errors in the receiver's side

This case is identical to the one without automatic acceptance (see section B.1.8) taking into account NOTE 1 and 17 in section B.2.19.

NOTE: The error messages that are mapped to 486 Busy Here are listed in Table 55.

Also on the path between the IM-ASs (the Messaging Server) similar errors could occur. In that case if the originating Messaging Server supports Store and Forward, it will behave in the same way and store the message.

B.2.9. Race conditions: Simultaneous INVITEs

Even if somewhat more unlikely in case of automatic acceptance, this case is identical to the one without auto-accept (see section B.1.9). NOTES 1, 2 and 17 in section B.2.19 apply as well.

B.2.10. Race conditions: New INVITE after a session is accepted

Even if somewhat more unlikely in case of automatic acceptance, this case is identical to the one without auto-accept (see section B.1.10). NOTES 1, 2 and 17 in section B.2.19 apply as well.

B.2.11. Store and forward: Message(s) displayed notifications via SIP MESSAGE with sender offline

This case is identical to the one without automatic acceptance (see section B.1.11). NOTES 2, 9, 10, 11 and 17 in section B.2.19 apply as well.

B.2.12. Interworking to SMS/MMS with automatic acceptance at the IWF

This case is identical to the one without automatic acceptance (see section B.1.12). NOTES 1, 2, 13, 17 and 18 in section B.2.19 apply as well.

B.2.13. Interworking to SMS/MMS with manual acceptance

This case is identical to the one without automatic acceptance (see section B.1.13). NOTES 1, 2, 13, 17 and 18 in section B.2.19 apply as well

B.2.14. Message Revoke: Successful Request

This case is identical to the one without automatic acceptance (see section B.1.14). NOTES 2, 7, 8 and 17 in section B.2.19 apply as well

B.2.15. Message Revoke: Failed Request

This case is identical to the one without automatic acceptance (see section B.1.15). NOTES 2, 7, 8 and 17 in section B.2.19 apply as well

B.2.16. Rejoining a Group Chat that timed out due to inactivity

This case is identical to the one without automatic acceptance (see section B.1.16). NOTES 2, 17 and 19 in section B.2.19 apply as well

B.2.17. Deliver Group Chat Messages while Chat is idle

This case is identical to the one without automatic acceptance (see section B.1.17). NOTES 2, 17, 19, 20 and 21 in section B.2.19 apply as well.

B.2.18. Race Condition: user rejoins active Group Chat which is torn down due to inactivity

This case is identical to the one without automatic acceptance (see section B.2.18). NOTES 2, 17, 19, 20 and 21 in section B.2.19 apply as well.

B.2.19. Chat and store and forward diagrams: Notes

Please note the following notes apply to diagrams in section B.2:

- NOTE 1 (B.2.1, B.2.2, B.2.3, B.2.4, B.2.5, B.2.8, B.2.9, B.2.10, B.2.12 and B.2.13): As said in section B.2, the inclusion of the message in the INVITE request is optional. If not included, the flows would be identical, but the message would be sent in the MSRP session instead as soon as it has been established.
- NOTE 2 (B.2.1, B.2.2, B.2.3, B.2.4, B.2.5, B.2.6, B.2.9, B.2.10, B.2.11, B.2.12, B.2.13, B.2.14, B.2.15, B.2.16, B.2.17 and B.2.18): 200 OK responses to SIP MESSAGE and MSRP SEND messages are omitted for clarity.
- NOTE 3 (B.2.3): In a multidevice scenario, if the device public GRUU in a delivery notification received from User B is different from the value for User A's device used in the ongoing MSRP session, a new session with automatic acceptance needs to be set up as specified in section 3.3.4.1.5.
- NOTE 4 (B.2.4): In a multidevice scenario, if the device public GRUU in a delivery notification received after the first INVITE is sent to User A is different from the value in the first one, a new SIP INVITE with the new device public GRUU needs to be sent towards A.

- NOTE 5 (B.2.3, B.2.4 and B.2.6): B could have to handle two incoming INVITEs, one from the Messaging Server on behalf of A to deliver messages and notifications that were stored to be forwarded, and a second one directly from A who happens to want to chat with B at the same time. B should recognize the INVITE from the Messaging Server on behalf of A and not tear it down when the new INVITE directly from A arrives: The INVITE from the Messaging Server has a Referred-By header and no isfocus tag, and the INVITE directly from A does not have a Referred-By header. Please note that the same applies to the case in which the order in which the INVITEs arrive is reversed.
- NOTE 6 (B.2.3, B.2.4, B.2.5 and B.2.6): The session established by the Messaging Server to deliver deferred messages to the destination only allows the receiver (client/device) to send back notifications (that is an INVITE with referred-by header will only allow message/imdn+xml in the CPIM part). If the user replies with a new message, then a separate session shall be established (That is if User B (the receiver) wants to reply, a new INVITE should be used) after all the deferred messages have been delivered.
- NOTE 7 (B.2.2, B.2.14 and B.2.15): In the diagram we have represented one of the possible mechanisms to detect that the user is not online (wait for the 480 response), however, there are alternative mechanisms (triggers, 3rd party registration) that can be also used by the Messaging Server for the purpose.
- NOTE 8 (B.2.3, B.2.4, B.2.5, B.2.14 and B.2.15): Note that in the scenario where the MSRP socket is closed between the Messaging Server and the Terminating client (B) in a deferred message delivery (due for instance to a small connectivity loss with the PDP context remaining active) and no re-registration takes place, if there are notifications pending (delivery or displayed) and all the deferred messages have been sent to B already (no need to open a new MSRP session), SIP MESSAGE can be used to confirm the pending delivery/display notifications that could not be sent over MSRP.
- NOTE 9 (B.2.11): Note that the deferred delivery of the display notifications stored to be forwarded in the Messaging Server will be performed as shown in section B.2.6.
- NOTE 10 (B.2.11): In the absence of a Messaging Server (neither in the sender's nor in the receiver's domain) and in the case the display notification fail to be delivered because the sender is offline, these notifications will be discarded and the receiver's client does not need to retry sending them. In any case, the next time User A manages to establish a chat session with User B, all the previous messages pending to receive the displayed notification will be marked as displayed/read.
- NOTE 11 (B.2.7 and B.2.11): In those scenarios where a Messaging Server is not available, neither in the sender's nor in the receiver's network, there is a chance that display notifications carried via SIP MESSAGE may be lost if the original sending client is offline when the receiver sends those display notifications (that is the last three messages in the diagram). To overcome this limitation, a terminal or client implementation should mark all the previous messages as displayed when a new chat message is received from the receiving user.
- NOTE 12 (B.2.3, B.2.4, B.2.5 and B.2.6): The session established by the Messaging Server to deliver deferred messages or notifications should be terminated once the all the messages and notifications have been delivered. In more detail:

  - When delivering deferred messages, the session should be terminated (by sending a BYE) either (whatever is shorter) when the display notification corresponding to the last deferred message has been received by the Messaging Server or, after a timer started on the reception of the delivered notification for the last message expires. This timer is defined by the Service Provider.

- NOTE 13 (B.2.12 and B.2.13): The predefined text for accepting and leaving a session is included for illustration purposes only as it is up to the Service Provider providing the interworking to configure an appropriate an appropriate text and announce that to the SMS/MMS user when appropriate.
- NOTE 14 (B.2.13): If the SMS (or MMS) user does not respond in time, the INVITE will have timed out and the used MSISDN may even be assigned to another session. For that reason the Messaging Server should check whether the SMS (or MMS) message comes from a user that is invited to the related session and if that is not the case or the MSISDN is not assigned to any session, a message is sent back informing the user that he cannot join the session any longer.
- NOTE 15 (B.2.4, B.2.5 and B.2.6): Whether a Messaging Server sets up a session for the delivery of notifications or sends them using SIP MESSAGE requests is up to its local policy. This could depend on factors such as the number of notifications that were stored or the number of messages for which notifications can be expected (during delivery of stored messages for instance).
- NOTE 16 (B.2.1, B.2.3, B.2.4 and B.2.5): When there is automatic acceptance and the first message is carried in the initial SIP INVITE, the delivery notification may be either delivered using a SIP MESSAGE or MSRP SEND leaving the choice up to the client implementation. In the diagrams we shown before, we have followed the second option.
- NOTE 17 (B.2.1, B.2.2, B.2.3, B.2.4, B.2.5, B.2.6, B.2.9, B.2.10, B.2.11, B.2.12, B.2.13, B.2.14, B.2.15, B.2.16, B.2.17 and B.2.18): As per [RFC5438], the message-id is conveyed in the messages via the imdn.Message-ID header and in the notifications via the value of the <message-id> element in the body of the IMDN.
- NOTE 18 (B.2.12 and B.2.13): The flows show interworking with SMS, but the flows in the SIP/MSRP part of the figure also apply when interworking with MMS.
- NOTE 19 (B.2.16 and B.2.17): As per sections 3.4.4.1.7 and 3.4.4.3.4.
- NOTE 20 (B.2.17 and B.2.18): The flow shows the Participating Function restarting the session before attempting the delivery. This is an implementation option to ensure that a session is established when the user sends content. The Participating Function may also choose to establish this session in parallel or only when there is actual content to be sent in the Chat.
- NOTE 21 (B.2.17 and B.2.18): The flow assumes that no display notifications were requested.

B.3. RCS Chat and multidevice

B.3.1. Delivery prior to acceptance

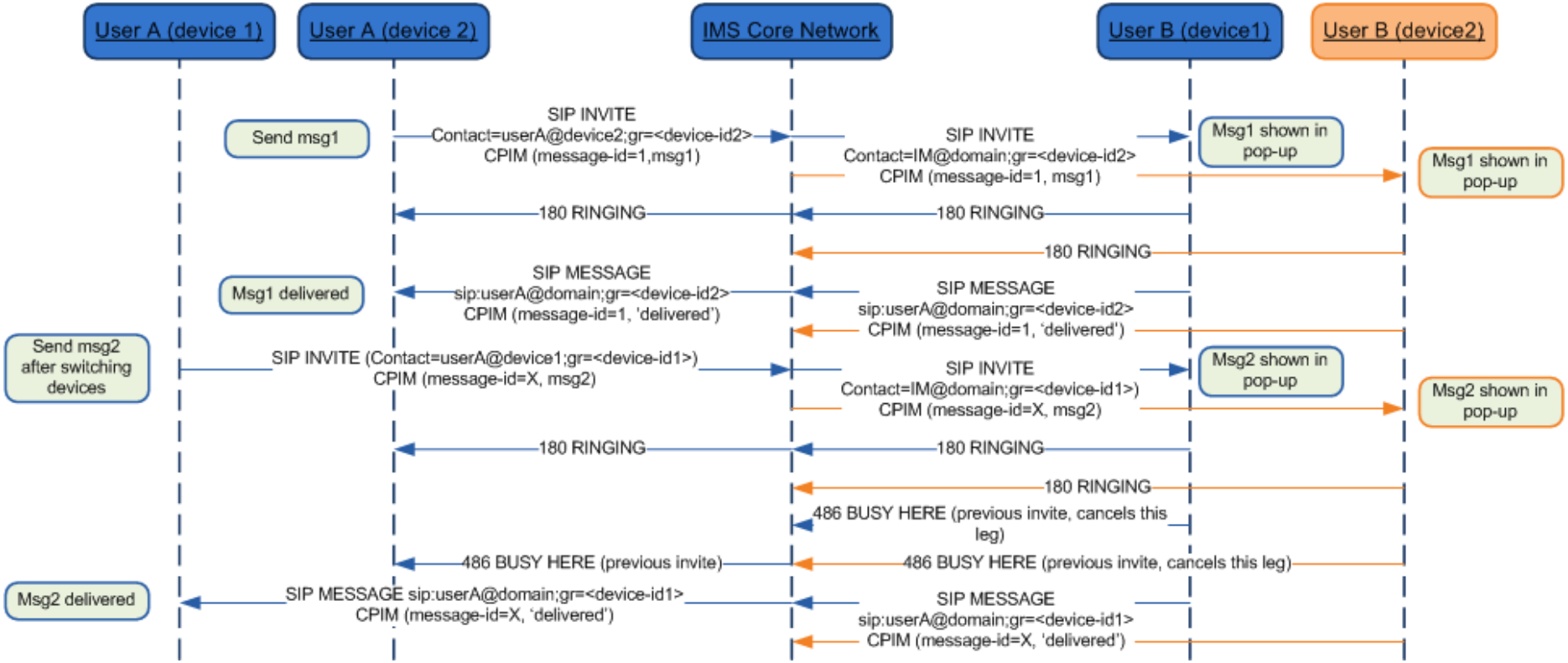


Figure 174: Chat and multidevice: Delivery prior to acceptance*

*: Check NOTES 1, 2, 3, 4 and 7 in section B.3.4

B.3.2. Post-acceptance behaviour

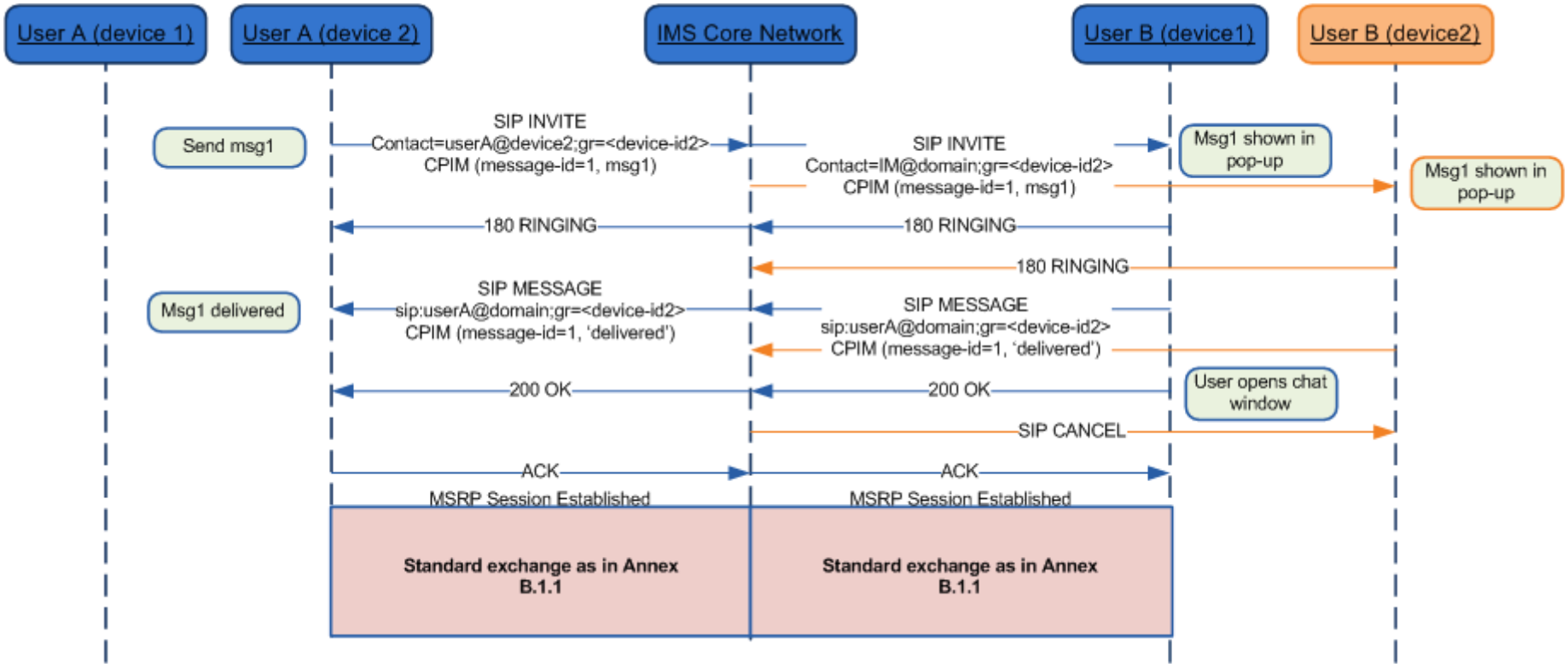


Figure 175: Chat and multidevice: Post-acceptance behaviour*

*: Check NOTES 1, 2, 3, 4 and 7 in section B.3.4

B.3.3. Behaviour with automatic acceptance

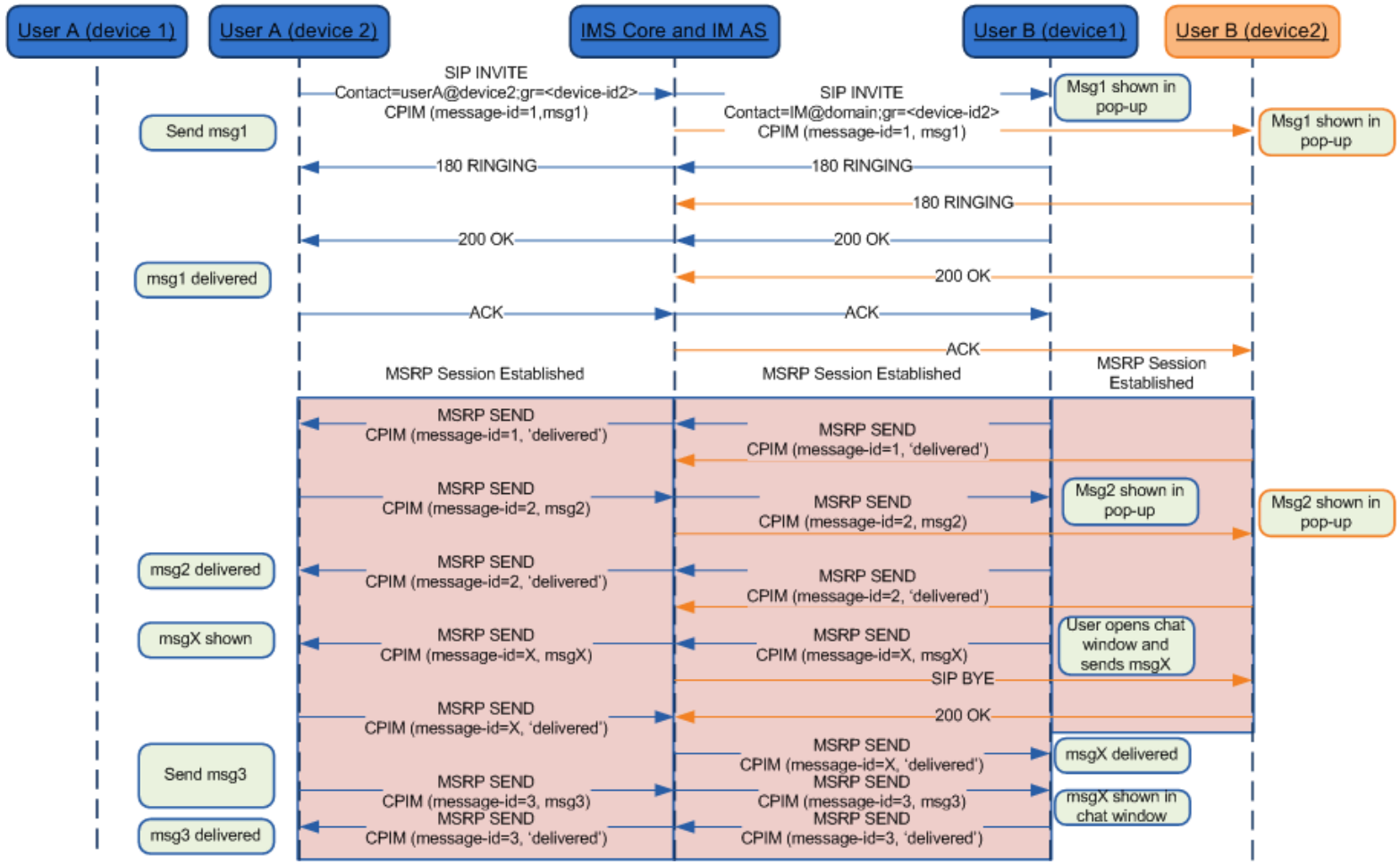


Figure 176: Chat and multidevice: Automatic acceptance*

*: Check NOTES 1, 2, 3, 4, 5, 6 and 7 in section B.3.4

B.3.4. RCS Chat and multidevice: Notes

Please note the following notes apply to diagrams in section B.3:

- NOTE 1 (B.3.1, B.3.2 and B.3.3): 200 OK responses to SIP MESSAGE and MSRP SEND messages are omitted for clarity.
- NOTE 2 (B.3.1, B.3.2 and B.3.3): As mentioned in section 2.11.3, the diagrams display the solution in a network supporting the pub-gruu generation. For a network supporting the sip.instance tag only, they would be equivalent with only a change of the mechanism to carry the device identifier (sip.instance instead pub-gruu).
- NOTE 3 (B.3.1, B.3.2 and B.3.3): The diagrams show that “delivered” notifications for messages for which such a notification was sent already, are suppressed by the network. As this cannot always be guaranteed, clients shall be prepared to receive such duplicate notifications and discard them silently. This holds also for display notifications and for notifications related to messages that were not sent by that client.
- NOTE 4 (B.3.1, B.3.2 and B.3.3): The SIP URIs in the diagrams (including those in the contact headers and Request URIs) are shown for illustrative purposes only. Any part of those URIs may thus differ in actual deployments. The details of the URIs are also dependent on the exact location in the network where the message is sent.
- NOTE 5 (B.3.3): The inclusion of the message in the SIP INVITE request is optional, if not supported, the message will be sent in the MSRP session as soon as that is established.
- NOTE 6 (B.3.3): To support this case forking in the terminating side needs to be done at the Messaging Server using the mechanisms defined in section 2.11.2 as forking in the IMS core will lead to a race condition.
- NOTE 7 (B.3.1, B.3.2 and B.3.3): As per [RFC5438], the message-id is conveyed in the messages via the imdn.Message-ID header and in the notifications via the value of the <message-id> element in the body of the IMDN.

B.4. Common Message Store Interaction: IMAP Flows (informative)

B.4.1. Summary of Use Cases

The following use cases are covered:

- Use Case 1: Device gains connectivity and checks for new content
- Use Case 2: Device fetches all objects related to a specific conversation
- Use Case 3: Device stores an SMS
- Use Case 4: Device deletes a specific conversation
- Use Case 5: Device saves a message to archive

For the purpose of the Use Case examples described in this annex, the following assumptions are made for the initial view of the Common Message Store:

RCS user A has several conversation folders under the default system folder

- Each Conversation folder contains all objects related to well-known criteria,
  - e.g. “TEL-URI-of-B”, or “Conversation-ID-for-GC-ABCD”
- The “TEL-URI-of-B” folder contains:
  - 1 session for a 1-to-1 chat between A&B, with 3 objects:
    - the 1st message from the INVITE sent by this user
    - 1 received reply message
    - The Session Info Object (SIO) for this conversation[36]
- The “TEL-URI-of-C” folder contains:
  - 2 standalone (pager mode) interworked SMS messages
    - 1 sent to C, 1 received from C
  - 2 new 1-to-1 chat messages from C:
    - the 1st messages from the two INVITEs received (which were not answered due to being offline)
    - An SIO for this conversation[36]
- The “INBOX/Conversation-ID-for-GC-ABCD/Contribution-ID-for-GC-ABCD” session history folder contains:
  - 1 SIO for the GC containing subject “Lunch?”
    - the GC was started by this user, between A, B, C, & D
  - 1 message sent by this user, for which user had requested a display report
  - 1 message received by this user from B
  - 1 Group State Object (GSO), because the Group Chat has ended, containing the participant list at the end of the Group Chat
  - 1 new IMDN (display-report) from D, received as a standalone message
    - Received while device was offline.

[36] the SIO only needs to be stored once per conversation since it provides information not otherwise in the MSRP chat messages

- Note: As per OMA CPM, the Group Chat session history folder is labelled with the Contribution-ID. As defined in this specification Contribution-ID has the same value as the Conversation-ID. Hence both session history sub folder and conversation folder have the same name.

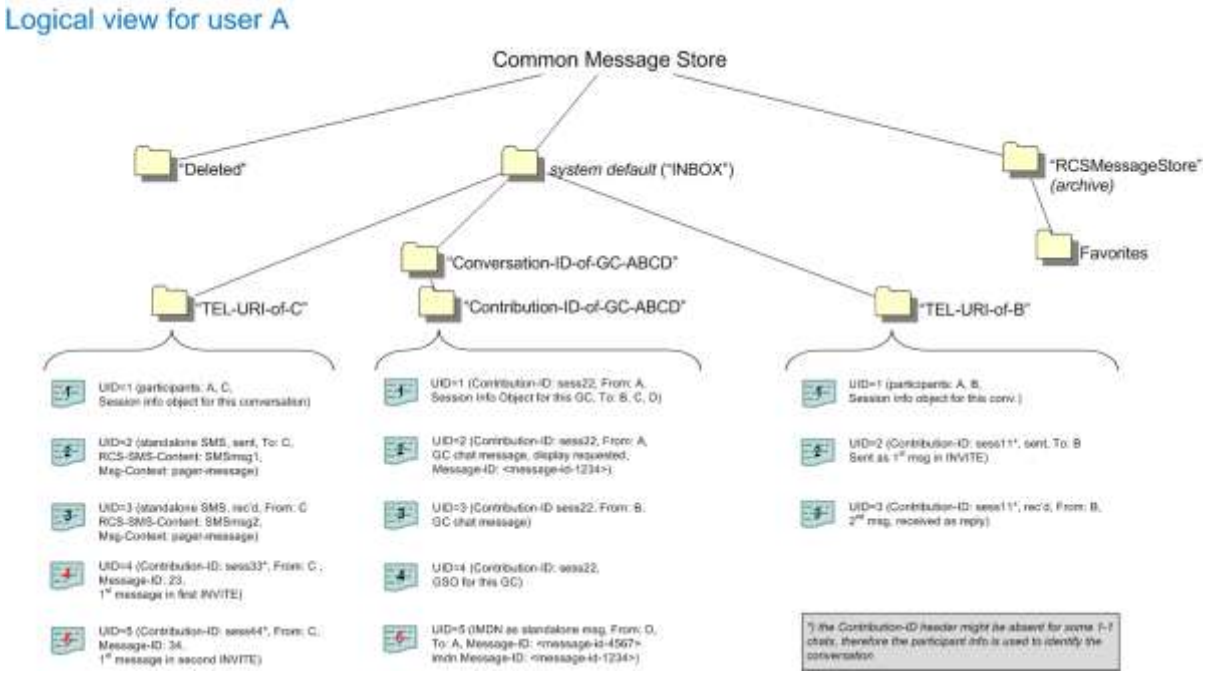


Figure 177: Initial Common Message Store view for IMAP synchronization flows

B.4.2. Use Case 1: Device gains connectivity and checks for new content

- Preconditions:
  - Initial Common Message Store view, with 3 new objects
  - Device was off and now needs to do a complete re-synchronization
  - The “Deleted” folder is empty
- High level flow:
  - Connect, Start TLS, Login
    - The Client connects to the server using the URL configured in the MESSAGE STORE URL attribute, starts the TLS layer, checks capabilities, and logs into the user’s Message Store account using the MESSAGE STORE USER and PASSWORD configuration attributes
  - List folders (with LIST-STATUS extension RFC 5819)
    - NEXTUID and UIDVALIDITY will indicate if anything new has arrived
    - If NEXTUID is higher and UIDVALIDITY is the same, new messages have arrived and the folder needs to be synchronized (Use Case 2)
    - If UIDVALIDITY has changed the folder needs to be synchronized (Use Case 2)
    - If number of messages is lower than what the client has stored, then messages have been deleted and the folder needs to be synchronized (Use Case 2)
    - In this case, the device notices a new folder and new messages

- Since the NEXTUID value for “INBOX/TEL-URI-of-C” and the “INBOX/Conv-with-GC-ABCD/Conv-with-GC-ABCD” folder is different from what client has stored, a synchronization is needed on both folders
  - For each folder:
    - Select the folder
    - Fetch flags and MIME headers for all UIDs newer than the last remembered UID
    - Compare meta information, e.g. Message-IDs for 1-to-1 Chat, with messages received via deferred delivery over SIP
    - Discover new objects (not also received over SIP)
    - Fetch each relevant new object completely
    - Arrange them in appropriate conversational views
    - Apply any IMDNs received
    - Close the folder
  - Log out, disconnect

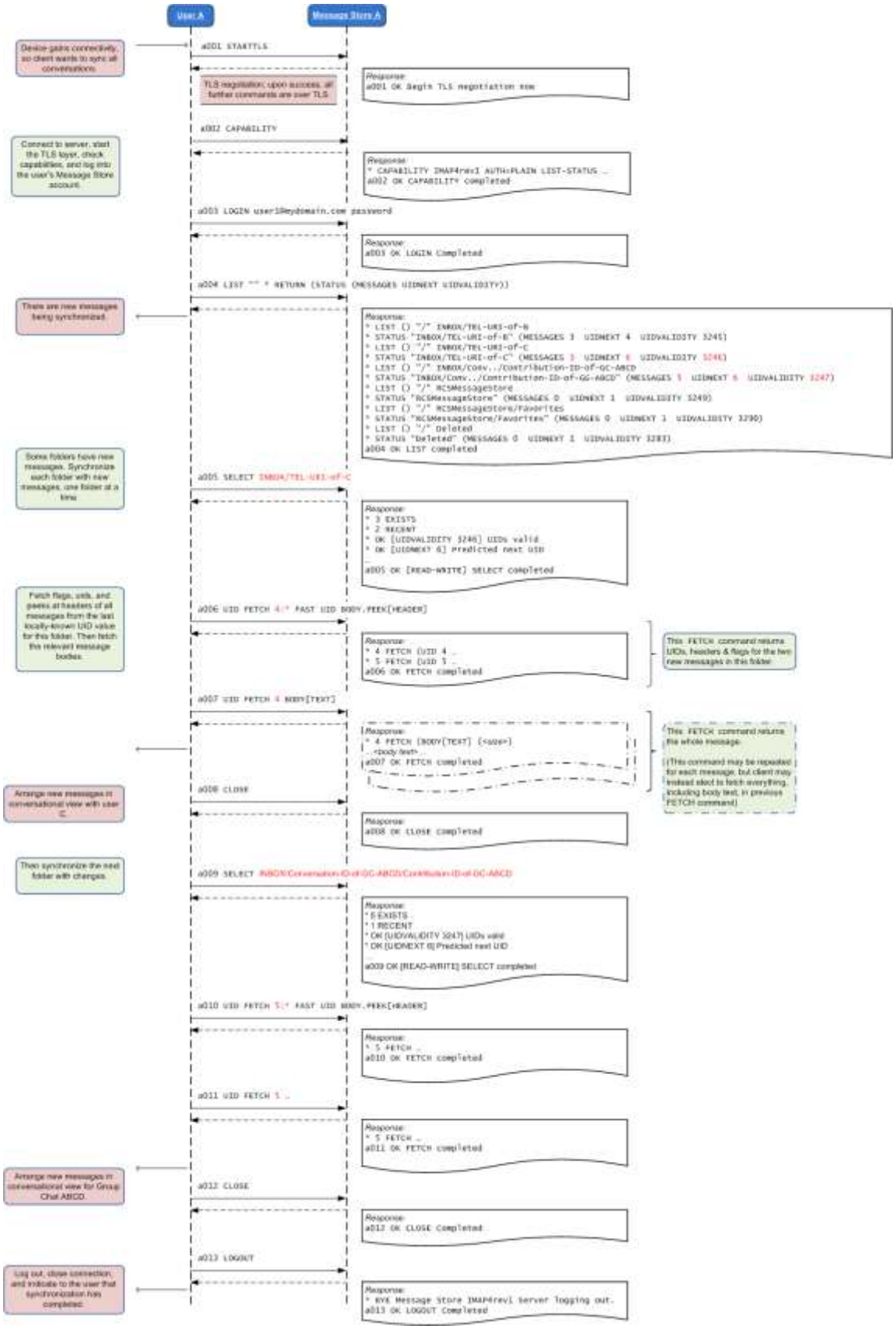


Figure 178: Use Case 1: Device gains connectivity and checks for new content

B.4.3. Use Case 2: Device fetches all objects related to a specific conversation

- Preconditions:
  - Same initial Common Message Store view, with 3 new objects
  - Device knows that this conversation is between users A and B, and it knows the conversation folder name
    - Normally, it is not necessary to fetch all objects related to a conversation from the Message Store Server
- High level flow:
  - Connect, Start TLS, Login
  - Select the appropriate conversation folder, e.g. “INBOX/TEL-URI-of-B”
  - Fetch all objects in this folder
    - Arrange all items in conversational view
    - Apply any IMDNs discovered in conversational view
    - Note which UIDs are missing from local storage view, consider those objects as deleted
    - Optionally, select the “Deleted” folder and search for objects which match criteria
  - Log out, disconnect

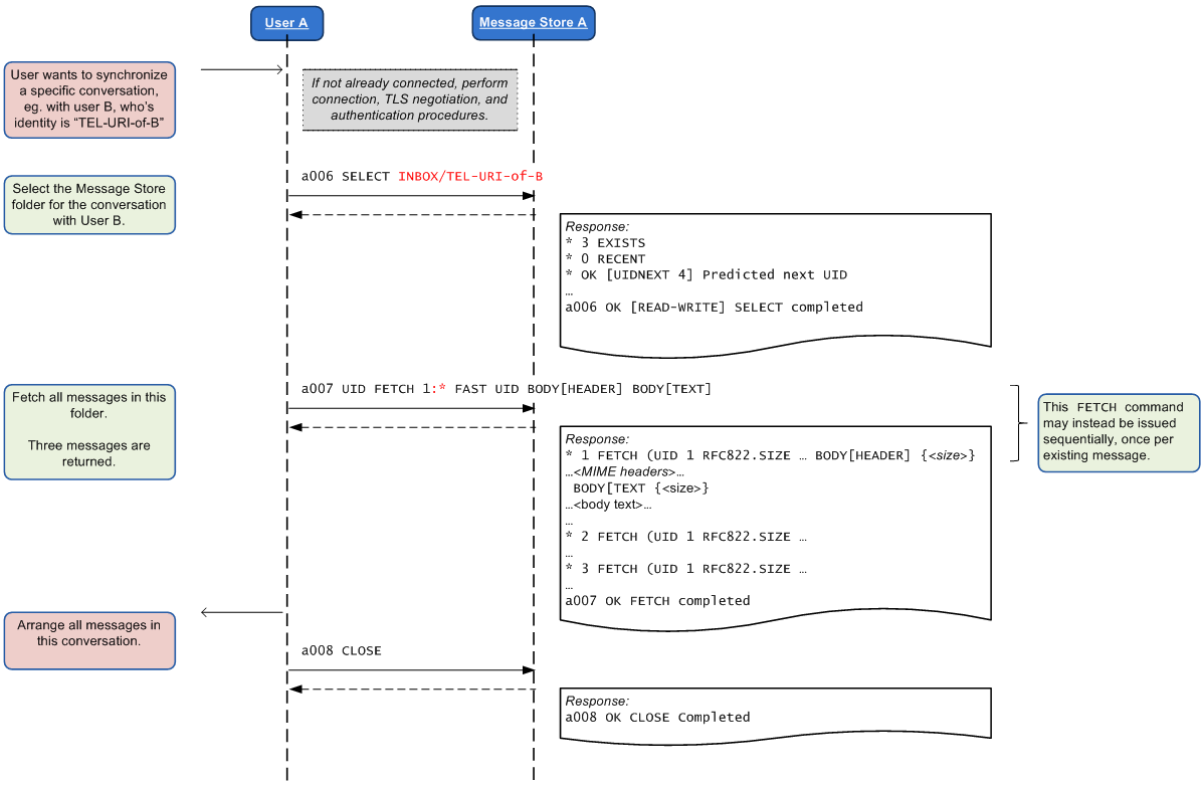


Figure 179: Use Case 2: Device fetches all objects related to a specific 1-to-1 chat conversation

B.4.4. Use Case 3: Device stores an SMS

- Preconditions:
  - Same initial Common Message Store view, with 3 new objects

- Device knows that this conversation is between users A and B
  - Normally, it is not necessary to fetch all objects related to a conversation from the Message Store Server
- The device has not found any standalone messages which match the To/From/RCS-SMS-Content the received SMS
  - It has also performed either a complete synchronization (Use Case 1) or a conversational view synchronization (Use Case 2) thus it can be sure that the received SMS doesn't match anything in the Common Message Store either

- High level flow:
  - Connect, Start TLS, Login
  - Select the appropriate conversation folder, e.g. "INBOX/TEL-URI-of-B"
  - Append the SMS message to the folder
    - Message-Context header is set to "pager-message"
  - Log out, disconnect

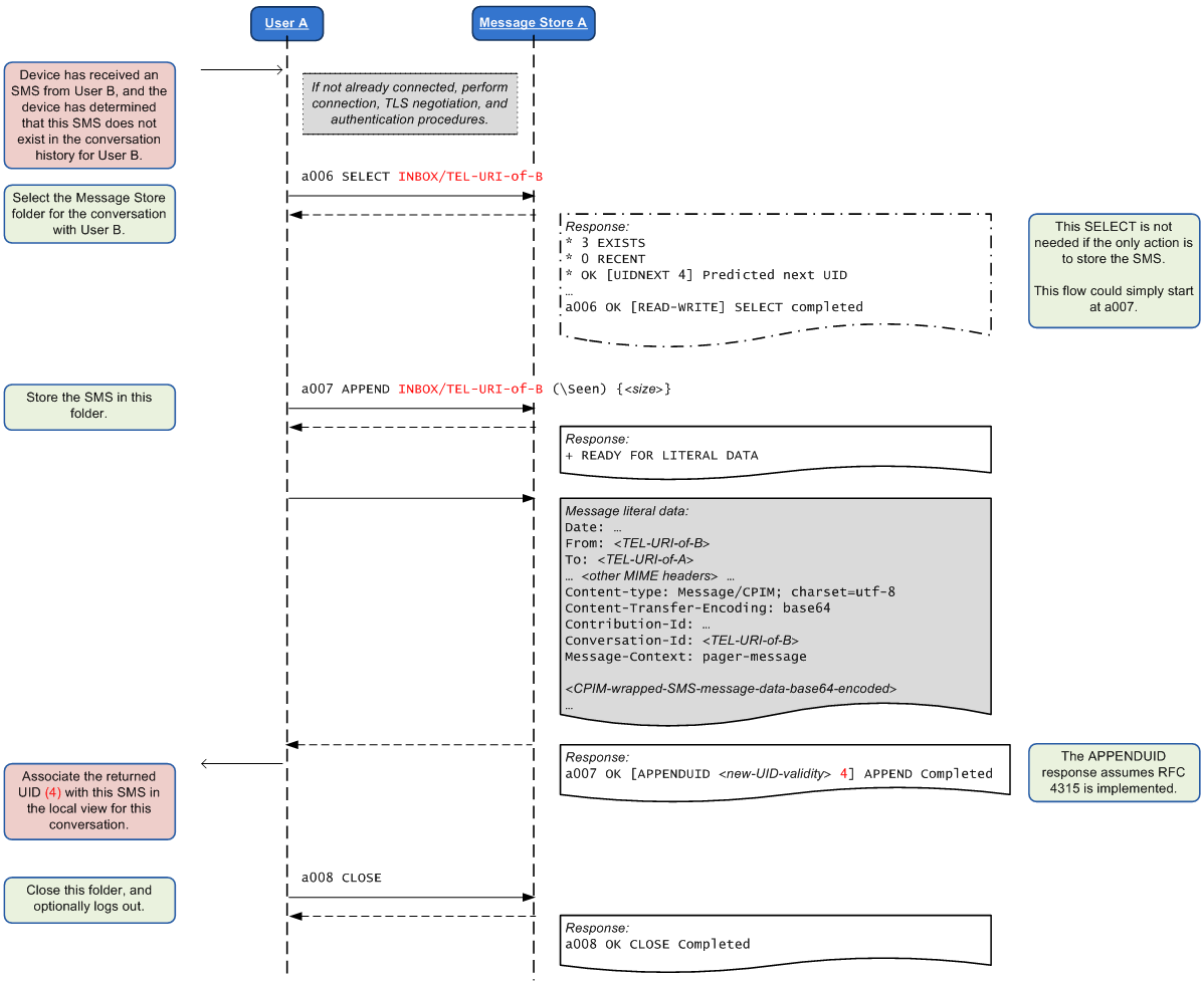


Figure 180: Use Case 3: Device stores an SMS

B.4.5. Use Case 4: Device deletes the conversation between A and B

- Preconditions:

  - Same initial Common Message Store view, all messages have been synchronized (Use Case 1)
  - Device knows that this conversation is between users A and B
    - Device knows the individual UIDs for each message and associated IMDNs between A and B
    - If it does NOT know the UIDs, the device must use Use Case 2 to first fetch all related messages and associated IMDNs
    - Normally, it is not necessary to fetch all objects related to a conversation from the Message Store Server
  - It is left to client implementation how to present deleted messages
- High level flow for Use Case 4a:
  - Connect, Start TLS, Login
  - Copy all objects of the appropriate conversation folder to the “Deleted” folder
  - Delete appropriate conversation folder, e.g. “INBOX/TEL-URI-of-B”
    - Assumes the Message Store server allows and supports deleting entire folder, and all contained messages inside are also deleted, not orphaned
  - Close, Log out, disconnect
- High level flow for Use Case 4b: If “undo” feature via Deleted folder is being used, then:
  - create the same-named folder under “Deleted”, copy messages to new folder, set the “\Deleted” flag on all messages in original folder, lastly expunge

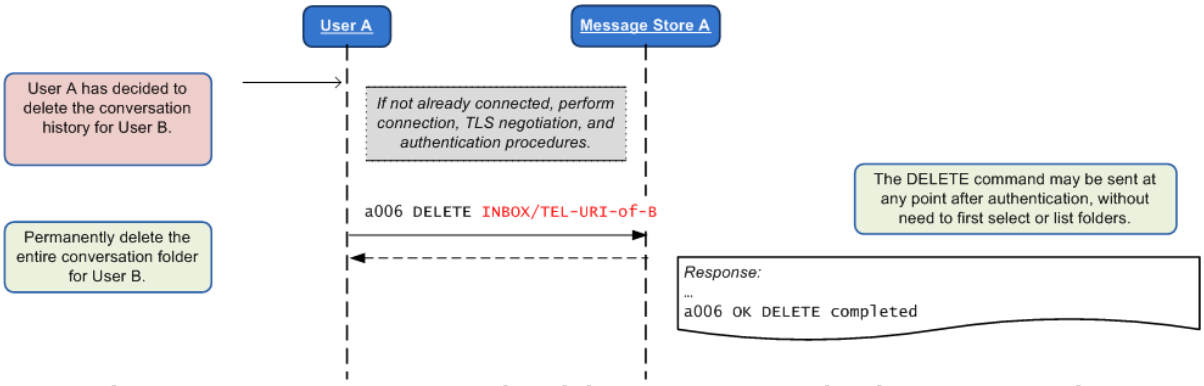


Figure 181: Use Case 4a: Device deletes a conversation between A and B

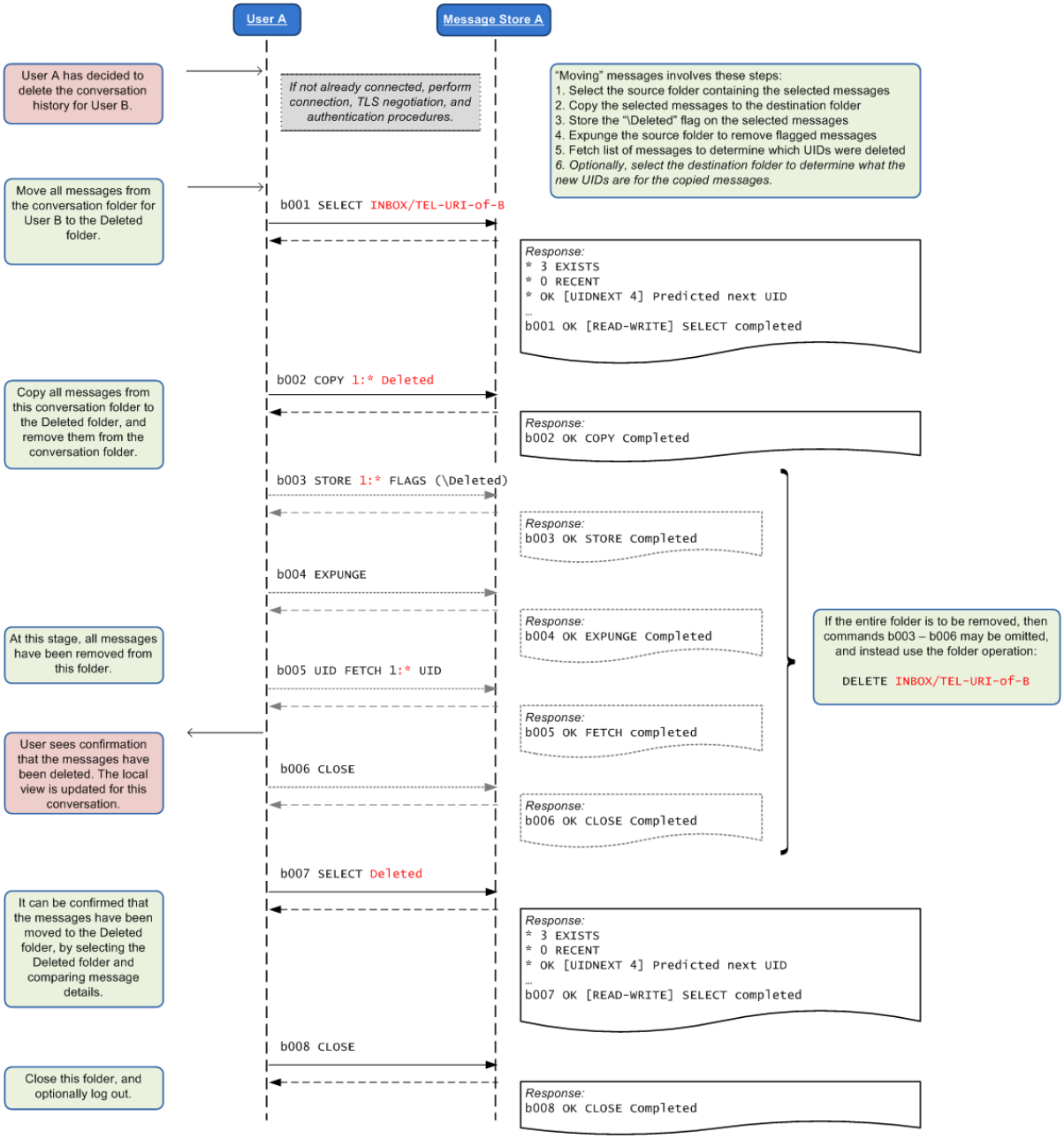


Figure 182: Use Case 4b: Device deletes a conversation between A and B by moving to a Deleted folder

B.4.6. Use Case 5: Device saves a message to archive

- Preconditions:
  - Same initial Common Message Store view, all messages have been synchronized (Use Case 1)
  - Device knows that this conversation is between users A and B
    - Device knows the individual UIDs for each message and associated IMDNs between A and B
    - If it does NOT know the UIDs, the device must use Use Case 2 to first fetch all related messages and associated IMDNs
    - Normally, it is not necessary to fetch all objects related to a conversation from the Message Store Server
  - It is left to client implementation how to present messages from the archive
- High level flow:
  - Connect, Start TLS, Login
  - Select the appropriate conversation folder, e.g. “INBOX/TEL-URI-of-B”
  - Copy the specific message with specific UID to the archive folder
    - E.g. “RCSMessageStore/Favourites” folder
  - Close, Log out, disconnect

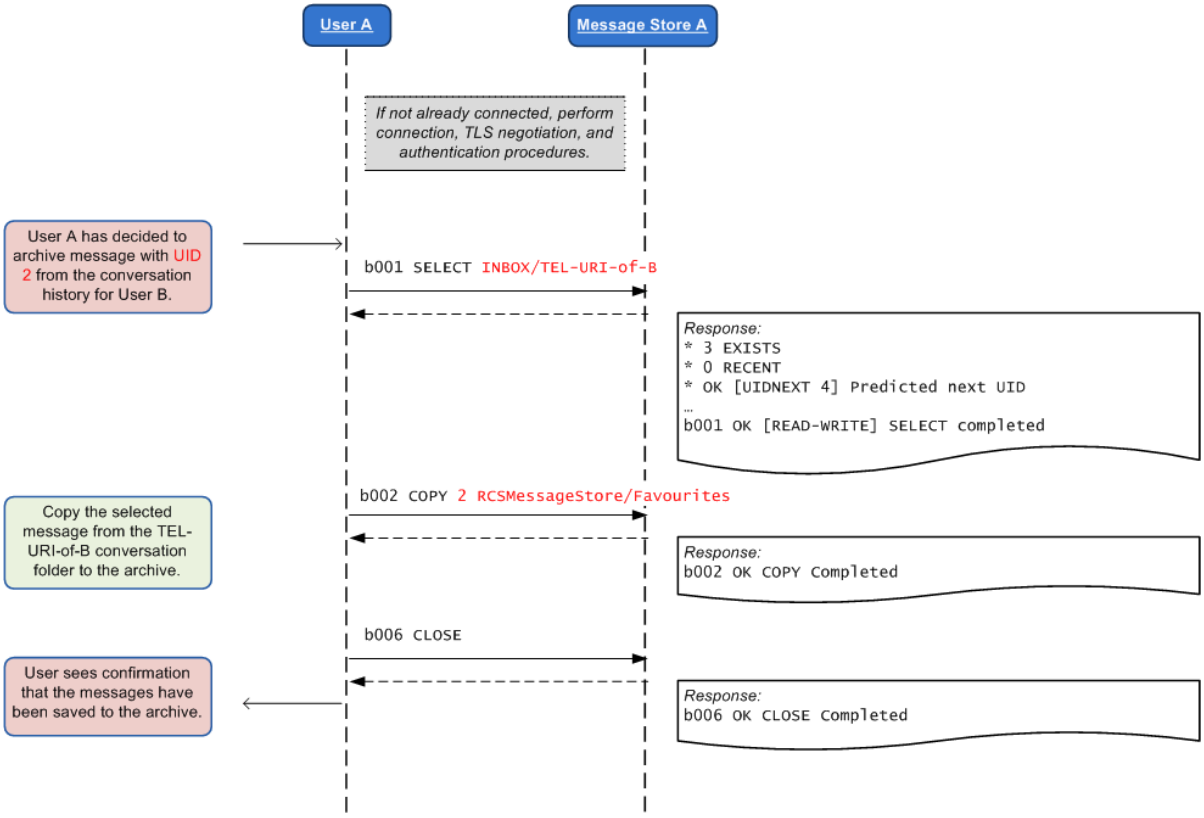


Figure 183: Use Case 5: Device saves a message to the RCSMessageStore archive

Annex C: Special Procedures

C.1. SIP/TCP and NAT traversal

As specified in section 2.8 when using SIP over TCP (or TLS), the client shall rely on the CRLF mechanism defined in [RFC6223]. However [RFC6223] does not provide the means to negotiate the direction in which these keep-alive requests are sent (it's always the party that initiated the SIP request that has to send keep-alive requests) and a device OS's scheduling policy may not always allow the client to meet the timing requirements for sending keep-alive requests. To overcome these limitations for clients running on such platforms a mechanism is provided in this annex which is also specified in an internet draft that has been submitted to the IETF (see [IETF-DRAFT-RKEEP]). This mechanism allows these clients to request to reverse the direction in which the keep-alive requests are sent (that is they will be sent from network to client) by including an 'rkeep' parameter in the Via header of the SIP request that is used in the same way as the 'keep' parameter defined in [RFC6223].

Like the server in [RFC6223], the client may include a proposed frequency (in seconds) of the keep-alive period by adding a value to the '*rkeep'* parameter (e.g. "rkeep=600"). This frequency shall not be set to a value smaller than 30 seconds. An Edge Proxy supporting this mechanism that receives requests that contain an 'rkeep' parameter in the top-most Via header can provide the following responses:

- If the *rkeep* value is provided by the client (e.g. rkeep=600) and it is acceptable according to the service provider policies, the registration response shall include the '*rkeep*' parameter in the top-most Via header when sending a reliable response on that request and shall remove the value (i.e. *rkeep* is sent back without a value).
- If the rkeep value is provided by the client but it is not acceptable based on the Service Provider policies, the Edge proxy shall include the '*rkeep*' parameter in the top-most Via header when sending a reliable response on that request and shall set the value to a default one (i.e. rkeep=180 [assuming 180 is the default value]).
- If the rkeep value is not provided by the client (e.g. rkeep without an specified value), the Edge Proxy shall provide a frequency value by setting a default value to the 'rkeep' parameter in its response (i.e. rkeep=180 [assuming 180 is the default value]).

Then it shall send double CRLF "ping" requests as defined in [RFC5626] to the client thereby complying to the specified interval and considering the connection as failed when no single CRLF "pong" response is received within 10 seconds.

An Edge proxy not supporting this mechanism shall not modify the rkeep parameter included by the client. The fact the value introduced by the client is not modified by the Edge Proxy shall be interpreted by the client as the Edge Proxy does not support the network initiated keep alive. Please note that this approach guarantees backwards compatibility.

NOTE1: it is highly recommended that clients not experiencing such scheduling limitations use the standard 'keep' mechanism defined in [RFC6223] and send the keep-alive requests themselves. For those clients the implementation of this section is therefore optional.

NOTE2: Alternatively a Service Provider could decide to rely on client platform specific notification mechanisms

NOTE3: The requirement to extend the keep-alive procedures to support network-initiated keep-alives has been brought into the IETF for standardization (see [IETF-DRAFT-RKEEP]). The procedures here will be updated once that work is completed. In particular this standardization process should allow the

client to detect that the network does not support network-initiated keep-alives as described above.

C.2. Examples of single registration architectures

NOTE: the diagrams of the network in this section are just for representation purposes.

C.2.1. Multi-stack approach

In this approach, each individual service or subset of services uses its own IMS stack, meaning, there could be several stacks running in a single device sharing the same IMS identity. In other words, each client uses its own IMS stack.

An example of the mentioned architecture is provided in the following figure showing VoLTE and RCS in a handset:

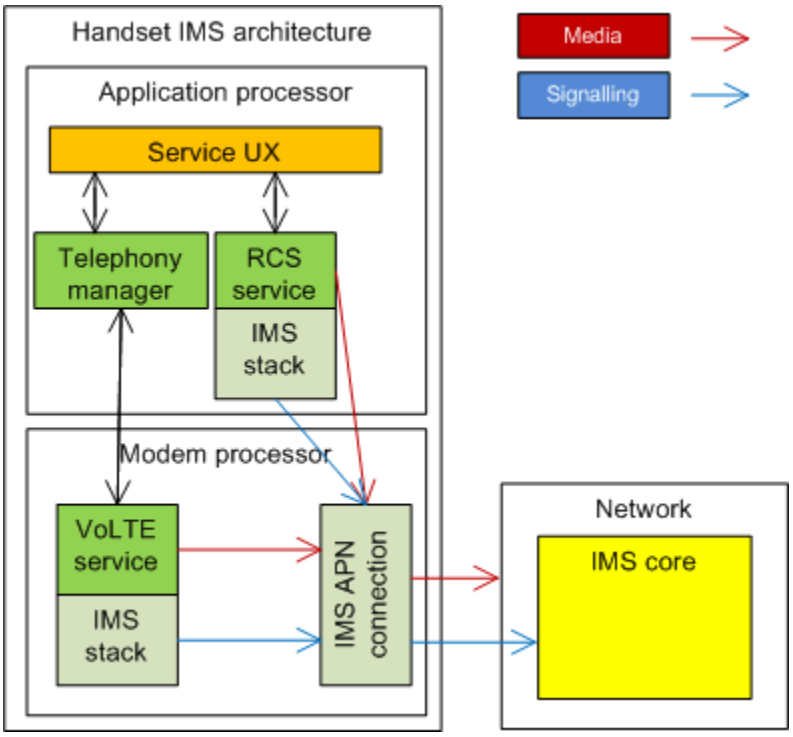


Figure 184: Multiple stack device architecture approach

Technically, this architecture is possible. The approach is to handle the different clients as separate instances of the same IMS registration as if they were running on different devices using the procedures described in section 2.4.2. The only consideration that shall be taken into account is that only one IMS stack can use the IMEI as sip.instance during registration as per [3GPP TS 24.229]. Therefore, only one RCS embedded client (as defined in section 2.2.2) is allowed per device.

This architecture is non-intrusive and relatively straightforward to implement because it does not require device integration. The drawback is however that several IMS registrations have to be maintained per device with the obvious impact in traffic and, more importantly, battery life. Consequently, this architecture is not recommended for mobile devices where battery life optimization is a key feature.

The recommended approach is therefore always that a device performs a single registration to the IMS core as it will consequently solve the issues presented above. The other

approaches discussed in the section C.2 provides examples of architectures that may be considered for that purpose.

C.2.2. IMS device API approach

In this approach, a single IMS stack is considered. In order to “share” the stack between clients a flexible IMS API is used. This API mainly covers signalling (SIP) and can optionally provide extensions to handle the media via the API mechanism as media can also be directly handled by the IMS clients. Figure 185 shows an application of this architecture, again, using the RCS and VoLTE services in a handset as IMS services examples

The benefits of this architecture resides in the fact that each device has only a single IMS stack and that therefore the IMS stack itself will be compliant to [3GPP TS 24.229] regarding the use of IMEI for sip.instance during registration and, more importantly, that it is more suitable for mobile devices with challenging battery life and connectivity considerations.

However, in order to make this approach successful, an activity to standardize a sufficiently flexible IMS API is required allowing all devices to share the same API principles and, respecting the OS and device API diversity, the syntax as much as possible.

Please note that the definition of this API is out of the scope of this document.

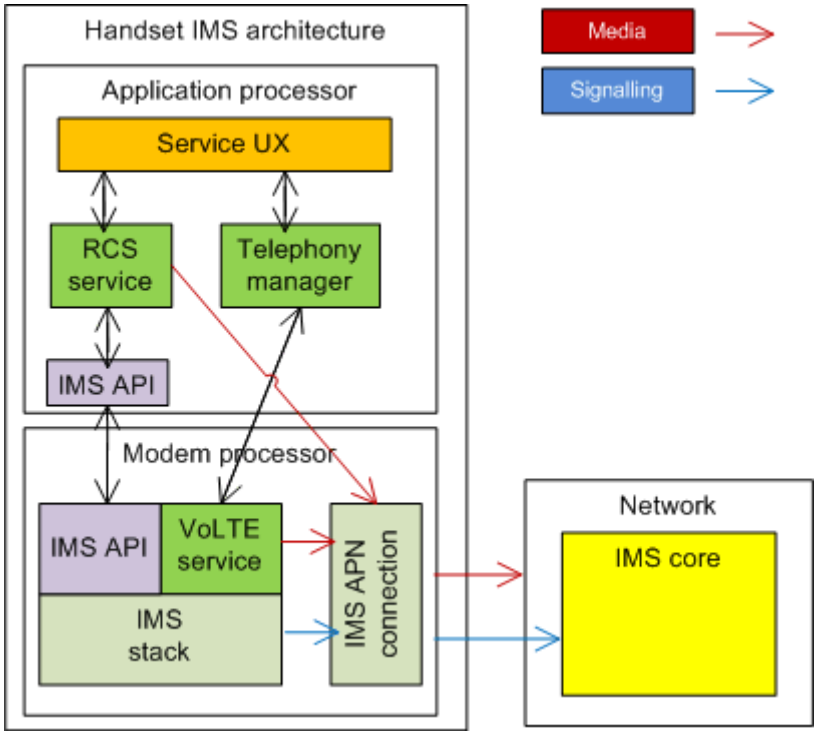


Figure 185: IMS API – single device stack architecture approach

C.2.3. IMS device with SIP back-to-back user agent and proxy approach

In this approach, multiple IMS stacks (one per client) are considered however only a single stack is actually registered with the IMS core (single registration). The idea is that a new architectural function, a SIP B2BUA+Proxy, is included in the device. The main responsibility of this function is the handling of the registration (including authentication and signalling security) with the IMS core which is the most resource-consuming activity from the battery management point of view. This new function acts in a transparent manner towards the different IMS client/stack pairs running on the device. It intercepts the registration and connectivity management SIP traffic (e.g. keep-alives) and manages it so that a single registration and connection is maintained with the IMS core.

Again the different clients running on the device shall be handled as multiple instances of the same IMS registration as if they are running in different devices using the procedures described in sections C.2.1 and 2.4.2. The SIP B2BUA is able to identify the different clients. Their sip.instance/gruu values shall be maintained internally however and not shared with the network. Only the sip.instance based on the IMEI is used during registration with the network, as per [3GPP TS 24.229].

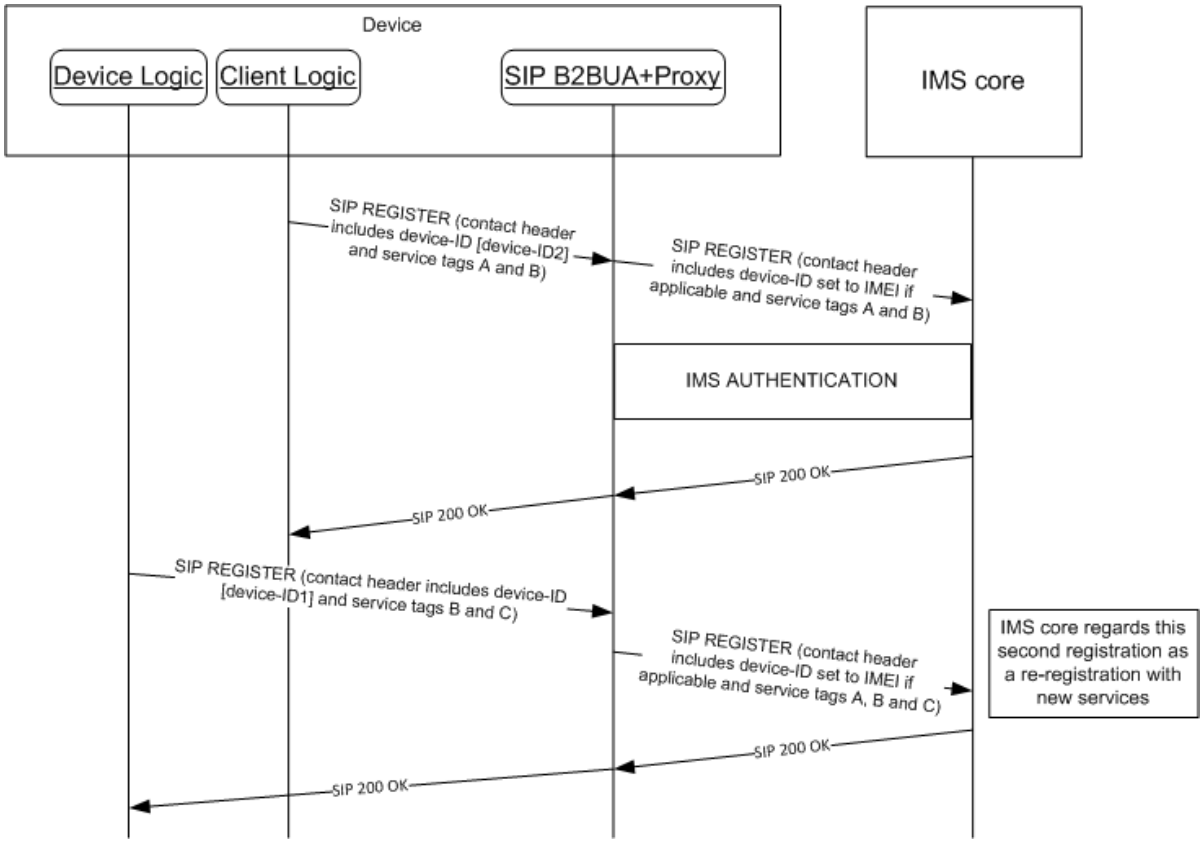


Figure 186: IMS device SIP B2BUA+Proxy stack registration procedure

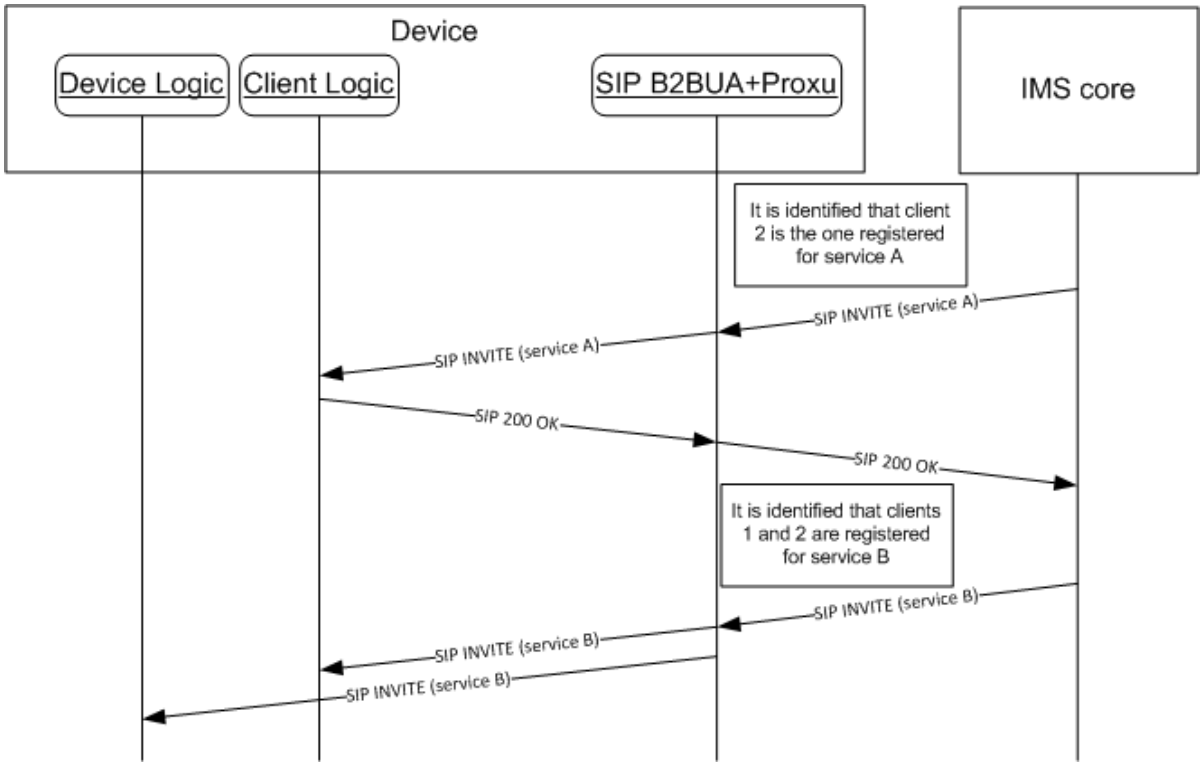


Figure 187: IMS device SIP B2BUA+Proxy stack SIP INVITE handling

The benefit of this architecture is that without giving up the idea of having one stack per IMS service, we are able to minimize the impact of such an approach, particularly on the battery life, a critical factor for mobile devices. The benefits come with the price that to make this architecture efficient, the new SIP B2BUA+Proxy function has to be deeply integrated in the OS, and ideally, in the hardware of the device.

Finally an example of this architecture is provided in the next figure again using RCS and VoLTE as the IMS services in a handset for representation purposes.

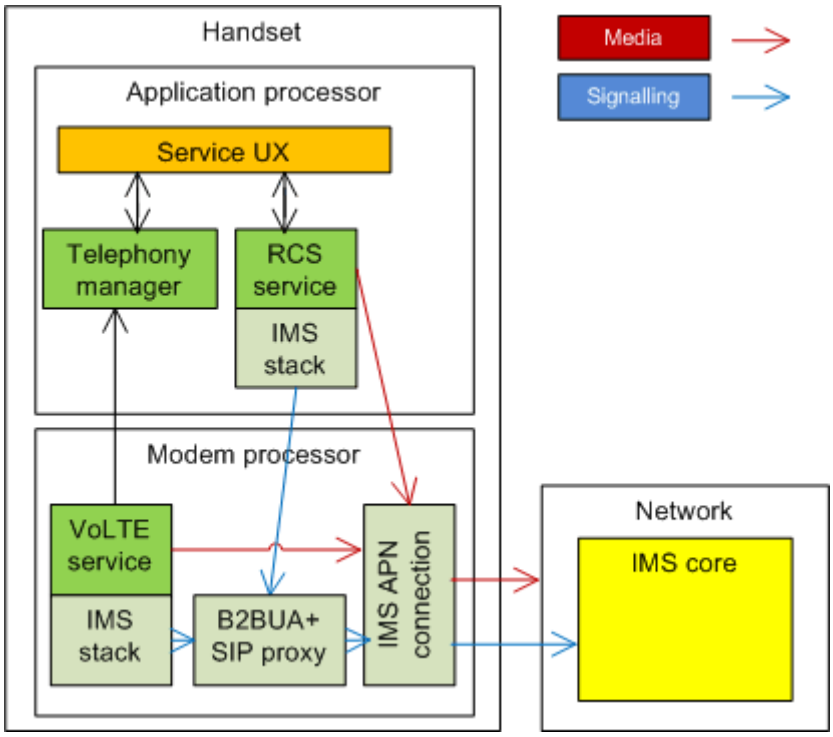


Figure 188: IMS device SIP B2BUA+Proxy stack architecture approach

C.3. Errata for RFC 5438

The following errata have been reported for [RFC5438] in [RFC5438Errata] and is important to be taken into consideration for RCS with respect to messaging and chat services:

- Errata ID: 3013
- Status: Held for Document Update
- Type: Technical
- Reported By: Dan Price
- Date Reported: 2011-11-04
- Held for Document Update by: Robert Sparks
- Section 7.2.1.1 says:

  From: Bob <im:bob@example.com>

  To: Alice <im:alice@example.com>

  NS: imdn <urn:ietf:params:imdn>

  imdn.Message-ID: d834jied93rf

  Content-type: message/imdn+xml

  Content-Disposition: notification

  Content-length: ...
- It should say:

  From: Bob <im:bob@example.com>

  To: Alice <im:alice@example.com>

  NS: imdn <urn:ietf:params:imdn>

imdn.Message-ID: d834jied93rf

Content-type: message/imdn+xml

Content-Disposition: notification

Content-length: ...

- Notes:

  None of the examples in this RFC (Request For Comments) comply with the format of CPIM defined in RFC 3862, in which the message metadata headers are separated from the headers of the encapsulated MIME object by a blank line.

C.4. Definition of RCS related MIME headers

NOTE: ABNF definitions of Contribution-ID and Conversation-ID can be found via [RCS5-CPM-MSGSTOR-ENDORS].

C.4.1. Definition of RCS-SMS-Content

The RCS-SMS-Content header is defined as an extension to the [RFC2822] field definitions. The limits for the occurrence of the field are defined in the following table:

Field	Min Number	Max Number
rcs-sms-content	0	1

Table 267: APN MO sub tree addition node

The field itself is defined in ABNF as follows:

```
rcs-sms-content   = "Rcs-Sms-Content:" sms-content-value CRLF
sms-content-value = ascii-value / non-ascii-value
ascii-value       = *160 (%x20-7E)
non-ascii-value   = "=?" charset "?" encoding "?" encoded-text "?="
                  ; encoding as defined in [RFC2047] for
                  ; encoded-word
charset           = "utf-8"
encoding          = "b"
```

NOTE: How to handle commands in the SMS messages is FFS.

C.5. Extension to Extension ICSI Release Version in User-Agent and Server headers

User-Agent and Server headers are used to indicate the release version and product information of the Extension to Extension Client.

The Extension to Extension Client shall implement the User-Agent and Server headers, according to the rules and procedures of [RFC3261] with the clarifications in this annex.

The User-Agent and Server headers ABNF are specified in [RFC3261] and extended as follows:

```
Server = "Server" HCOLON server-val *(LWS server-val)
User-Agent = "User-Agent" HCOLON server-val *(LWS server-val)
server-val = product / comment
product = ExttoExt-product / token [SLASH product-version]
product-version = token
```

C.5.1. Extension to Extension Version 1.0

This specification allows having several server-val tags. The first of those server-val tags shall be encoding according to the following ABNF:

```
ExttoExt-product = "ExttoExt-" ExttoExt-device-token (SLASH ExttoExt-product-version)
ExttoExt-device-token = "client" | "serv" token
ExttoExt-product-version = "Ext1.0"
```

Example:

In this example the Extension to Extension Client acting as UAC and the one acting as UAS are Extension to Extension release version 1.0 products. One of the Extension to Extension Clients has inserted its own company and product name and version "ABC-Extensions1000/v1.01".

```
User-Agent: ExttoExt-client/Ext1.0 ABC-Extensions1000/v1.01
Server: ExttoExt-serv/Ext1.0
```

Annex D: WebRTC and other ways to access the RCS/IMS network (Informative)

This UNI specification defines a standard way for a device/client to access the RCS functionality in the network to provide end-to-end RCS services. However there are other ways that are different from this UNI specification, that also allow a device/client to access the RCS functionality to provide end to end RCS services. These alternatives are not meant to replace or to modify this UNI specification but to complement it in different deployment environments.

The network architecture and device runtime environment described in this section is to support WebRTC access or UNI interfaces that can be used to expose RCS network functionality to 3rd party clients or operator provided clients (i.e. 1st party clients). Therefore this annex does not mandate any client behaviour for such applications.

NOTE: The text in this Annex is subject to ongoing standardization efforts in 3GPP, OMA, and IETF, and to be updated once the standards become available.

D.1. Introduction to WebRTC

Web Real-Time Communication (RTCWeb/WebRTC) is a suite of IETF and W3C standards (see [IETF-DRAFT-RTCWeb_Overview] and [W3C WebRTC]) that allows web browsers to run real-time media (containing audio, video, and data channels) in a peer-to-peer fashion from web browser to web browser, or between a web browser and a media gateway.

The W3C standards body has defined APIs for browsers to support the media functions of an IP communications client. The browsers support complex media layer handling functions including Codecs, Echo Cancellers, Jitter buffer, NAT traversal and Security functions to make it easier for web developers to integrate real-time communication services into websites and web applications.

WebRTC does not specify an end to end communications network architecture. A network-based architecture for the support of WebRTC client access to IMS is defined in [3GPP TS 23.228]. This specification defines, for example, how WebRTC clients access IMS, reusing IMS client security credentials and/or public identities/credentials as appropriate, the way IMS clients communicate with WebRTC clients connected to IMS and the ability to realise any IMS services to the WebRTC client.

The IETF RTCWeb specifications only partially specify the signalling. In particular, WebRTC requires SDP to be used to describe the media streams involved in the session, and the offer-answer model to negotiate the media but the signalling transport protocol is not specified by IETF.

The specification of the signalling protocol is necessary in order to design a complete WebRTC compatible IMS/RCS end to end architecture. Therefore, in the context of RCS, the following signalling architectures are recommended:

- Signalling using SIP over secure WebSocket as defined by 3GPP [3GPP TS 23.228] and summarized in section D.2.
- Signalling using RESTful NetAPI for WebRTC signalling as defined by OMA [REST WEBRTC SIG API] and summarized in section D.3

D.2. WebRTC RCS Clients using SIP over WebSocket Signalling

The WebRTC IMS architecture in 3GPP [3GPP TS 23.228] enables support of WebRTC clients (see Figure 189).

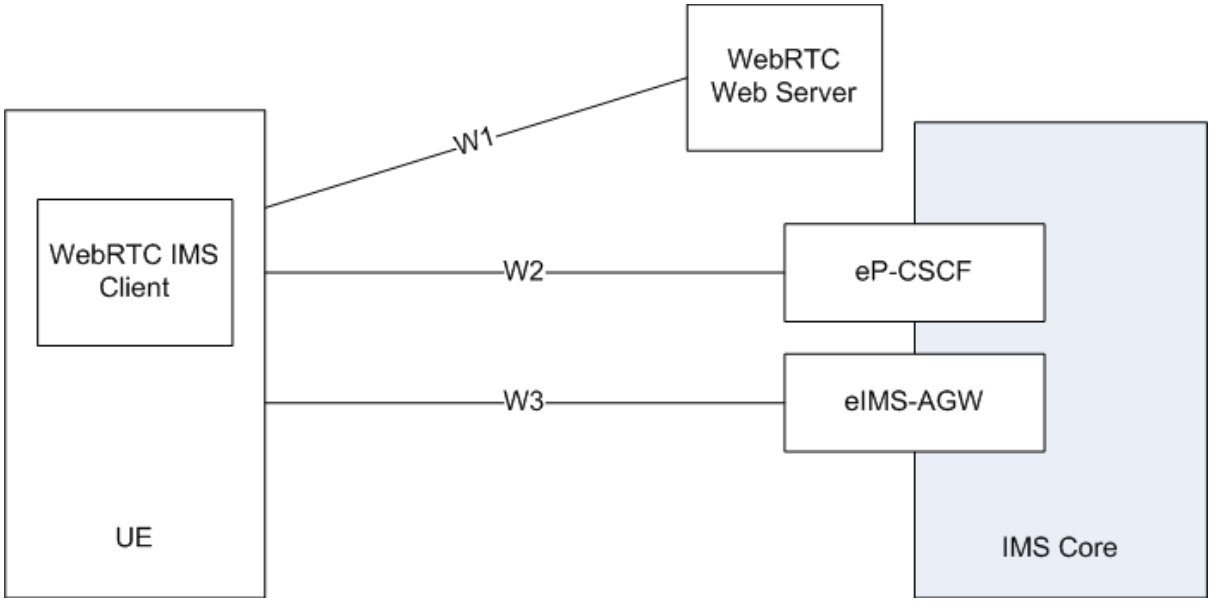


Figure 189: RCS architecture for WebRTC clients – SIP over WebSocket

The figure shows the architecture for RCS clients running in WebRTC enabled browser environments. This architecture supports:

- An interface W1 between the WebRTC IMS Client and the WebRTC Web Server, as described in [3GPP TS 23.228]. HTTPS is normally used to access the web page providing the user interface for the Client and to download the JavaScript application to the browser.
- A signalling plane interface W2 between the WebRTC IMS Client and the enhanced P-CSCF (eP-CSCF) and a Media plane interface W3 between the WebRTC IMS Client and the enhanced IMS-AGW (eIMS-AGW), reusing the UNI interface as specified in this specification but running on protocol stacks adequate for WebRTC clients as described in [3GPP TS 23.228] and summarized in Table 268.

Table 268 summarises the protocols used by WebRTC/SIP RCS clients to access the IMS Core.

Interface	Protocol name	Description	WebRTC RCS Client to gateway Transport layer	WebRTC RCS Client browser API
W2	SIP	Client-IMS core signalling protocol	WebSocket Protocol [RFC6455] SIP over websocket [RFC7118] TCP/IP	WebSocket API as defined in [W3C WS]
W3	MSRP	chat messages, media (pictures) and file exchange protocol	DataChannel transport as defined by [3GPP TS 23.228] U.1.5.1 (i.e. SCTP/DTLS/UDP/IP)	WebRTC API (Data channel control and data access) [W3C WebRTC]
	RTP	Real Time Media (voice and video) exchange	MediaStream track transport as defined by [3GPP TS 23.228] U.1.5.4. (i.e. SRTP/UDP/IP)	WebRTC API (control of real-time media) [W3C WebRTC]

Table 268: RCS protocols for WebRTC clients – SIP over WebSocket

D.3. Device/Clients using RESTful NetAPIs

This architecture option is based on the use of a gateway exposing RCS APIs. It provides following interfaces also summarized in Table 269 and Figure 190:

- An RCSAPI interface will use RESTful Network API for WebRTC Signalling [REST WEBRTC SIG API] and Network API for Notification Channel [REST RCS API] for signalling.
- Interface W3 will only be used for voice and video RTP media.
- The RCSAPI interface also supports other RCS services (e.g. File Transfer, Chat) based on RESTful NetAPI [REST RCS API].

NOTE: The deployment architecture for RCS API Gateway is not specified and open to vendor and operator deployment decision.

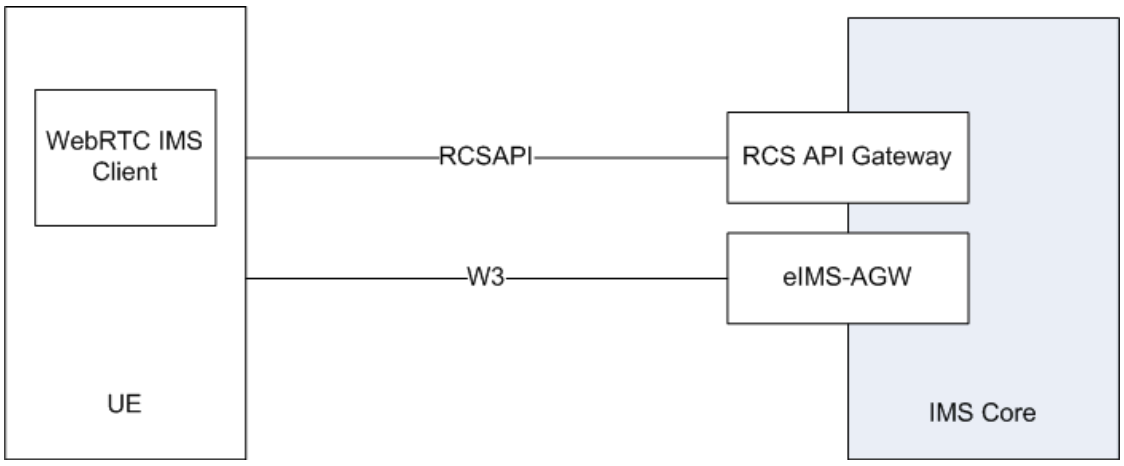


Figure 190: RCS architecture for WebRTC clients – RESTful NETAPI

The following table summarises the list of protocols employed by WebRTC RCS clients using RESTful NETAPIs for signalling and data exchanges.

Interface	Protocol name	Description	WebRTC RCS Client to gateway Transport layer	WebRTC RCS Client browser API
RCS API	RESTful Network API for WebRTC Signalling [REST WEBRTC SIG API]	Client/Gateway-IMS core signalling protocol	HTTP for Network APIs. WebSocket may be used only for notification transport as defined in RESTful Network API Notification Channel.	XMLHttpRequest API as defined in [W3C XHR] for Network APIs. WebSocket API as defined in [W3C WS] may be used only for notification transport as defined in RESTful Network API for Notification Channel.
	Other OMA RESTful NetAPI (e.g. Chat API, File Transfer API) [REST RCS API]	Chat messages, media (pictures) and file exchange protocol	HTTP for NetworkAPIs. WebSocket may be used only for notification transport as defined in RESTful Network API Notification Channel	XMLHttpRequest API as defined in [W3C XHR] for Network APIs. WebSocket API as defined in [W3C WS] may be used only for notification transport as defined in RESTful Network API for Notification Channel.
W3	RTP	Real Time Media (voice and video) exchange	MediaStream track transport as defined by [3GPP TS 23.228] U.1.5.4. (i.e. SRTP/UDP/IP)	WebRTC API (control of real-time media) [W3C WebRTC]

Table 269: RCS protocols for WebRTC clients – RESTful NETAPI

Document Management

Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
1.0	13 August 2012	First version for RCS 5.1 based on approved RCS 5.0 specification version 1.0 Approved by DAG and PSMC	PSMC	Tom Van Pelt / GSMA
1.0	26 September 2012	Added RCC.07 number		Tom Van Pelt / GSMA
2.0	02 May 2013	Applied MCR1001 approved by DAG and PSMC	PSMC	Tom Van Pelt / GSMA
3.0	25 September 2013	Applied MCR1002 approved by DAG and PSMC	PSMC	Tom Van Pelt / GSMA
4.0	28 November 2013	Applied MCR1003 approved by DQR and Global Specification Group (GSG)	GSG	Tom Van Pelt / GSMA
5.0	07 May 2014	First version of the document for RCS 5.2: Include approved CR1004	GSG	Tom Van Pelt / GSMA

Other Information

Type	Description
Document Owner	Network 2020 Programme, Global Specification Group
Editor / Company	Tom Van Pelt / GSMA

It is our intention to provide a quality product for your use. If you find any errors or omissions, please contact us with your comments. You may notify us at prd@gsma.com

Your comments or suggestions & questions are always welcome.