# Rich Communication Suite 9.0 Advanced Communications Services and Client Specification

## Version 10.0

## 06 December 2018

*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 Principles and Vision

RCS (Rich Communication Suite) provides a framework for discoverable and interoperable advanced communication services and detailed specifications for a basic set of such advanced communication services. The functional requirements for these services and the framework components as well as their integration in the client's User Experience (UX) are described in [PRD-RCC.71].

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).

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 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.
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. These same mechanisms can be used to initially discover the service capabilities of all the contacts within an address book when the user first registers for the service.

## 1.2 Scope

This document focuses mainly on the User Network Interface (UNI) which to a large extent also determines the Network-Network Interface (NNI). The interconnect-specific aspects of the NNI are described in a separate document (see [PRD-IR.90]). Also the functional requirements, the UX and client local aspects are described in a separate document (see [PRD-RCC.71]).

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 customisations and changes to optimise or differentiate their networks however.

It should be noted that all pictures and flow diagrams are for informative purposes only.

## 1.3 Definition of Terms

| Term | Description |
|---|---|
| 2G | 2nd Generation of Global System for Mobile Communications (GSM) |
| ACK | Acknowledgement |
| AF | Anonymization Function |
| ALG | Application Layer Gateway |
| AMR | Adaptive Multi-Rate |
| APN | Access Point Name |
| AP | Authentication Proxy |
| API | Application Programming Interface |
| AS | Application Server |
| AuC | Authentication Centre |
| BA | Broadband Access |
| B2BUA | Back-to-Back User Agent |
| bool | Boolean |
| bps | Bits per second (used with Mbps: Mega-, kbps: kilo-) |
| BSF | Bootstrapped Security Function |
| B-TID | Bootstrapping Transaction Identifier |
| CAB | Converged Address Book |
| CFS | Client Fallback to SMS |
| Chatbot | See [PRD-RCC.71] |
| Chatbot Information | meta information on a Chatbot, amongst others a description and contact information |
| Chatbot Information Function | A function providing meta information on a Chatbot. The provided information will include amongst others a description and contact information |
| Chatbot Platform | See [PRD-RCC.71] |
| CPIM | Common Profile for Instant Messaging |
| CPM | Converged IP Messaging |
| CRLF | Carriage Return Line Feed |
| CS | Circuit Switched |
| 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 |
| eIMS-AGW | Enhanced IP Multimedia Subsystem-Access GateWay |

| Term | Description |
|---|---|
| eP-CSCF | Enhanced Proxy-Call Session Control Function |
| EPSG | European Petroleum Survey Group |
| EUCR | End User Confirmation Request |
| FIFO | First IN First Out |
| FIR | Full Intra Request |
| FQDN | Fully Qualified Domain Name |
| FTTH | Fibre To The Home |
| GAA | Generic Authentication Architecture |
| GBA | Generic Bootstrapping Architecture |
| GBR | Guaranteed Bitrate |
| GML | Geography Markup Language |
| GPRS | General Packet Radio Service |
| Group Chat Session Identity | Public Service Identity assigned to a Group Chat. It is assigned by the Messaging Server and used by the client for addressing the Messaging Server. |
| GRUU | Globally Routable User agent URI |
| GSM | Global System for Mobile Communications |
| GSMA | GSM Association |
| GSO | Group State Object |
| HOS | Home Operator Services |
| HPLMN | Home Public Land Mobile Network |
| HSPA | High Speed Packet Access |
| HTTP | Hyper-Text Transfer Protocol<br>NOTE: 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. Whether for a particular request HTTP or HTTPS is used depends on the scheme specified in the definition of that request. |
| 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 |
| IEI | Information Element Identifier |
| IETF | Internet Engineering Task Force |
| IM | Instant Messaging. The term chat is also applied in this document to the same concept. |
| IMDN | Instant Message Disposition Notification |
| IMEI | International Mobile Station Equipment Identity |

| Term | Description |
|---|---|
| IMPI | Internet Protocol Multimedia Subsystem Private Identity |
| IMPU | Internet Protocol Multimedia Subsystem PUblic identity |
| IMS | Internet Protocol Multimedia Subsystem |
| 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 |
| JSON | JavaScript Object Notation |
| JWK | JSON Web Key |
| JWS | JSON Web Signature |
| KB | KiloByte (i.e. 1024 bytes) |
| kB | Kilobyte 1 kilobyte = $10^3$bytes = 1000bytes. |
| LSB | Least Significant Bit |
| LTE | Long Term Evolution |
| MaaP | Messaging as a Platform |
| MAP | Mobile Application Part |
| Messaging Server | A server providing support for the standalone messaging service (see section 3.2.2) and/or Chat and Group Chat (see sections 3.2.3 and 3.2.4) according to [RCS-CPM-CONVFUNC-ENDORS] |
| MIME | Multipurpose Internet Mail Extensions |
| MMS | Multimedia Message Service |
| MMS-C | Multimedia Messaging Service Centre |
| MMTEL | MultiMedia TELephony |
| MNO | Mobile Network Operator |
| MO | Management Object |
| 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 |
| NAL | Network Abstraction Layer |
| NAT | Network Address Translation |
| NFS | Network Fallback to SMS |
| NGBR | Non-Guaranteed Bitrate |

| Term | Description |
|---|---|
| NNI | Network Interface |
| NW | NetWork |
| OEM | Original Equipment Manufacturer |
| OMA | Open Mobile Alliance |
| OS | Operating System |
| P-CSCF | Proxy-Call Session Control Function |
| PC | Personal Computer |
| PCC | Personal Contact Card |
| PDP | Packet Data Protocol |
| PIDF | Presence Information Data Format |
| PKI | Public Key Infrastructure |
| Plug-in | See [PRD-RCC.71] |
| POSIX | Portable Operating System Interface |
| PPS | Picture Parameter Set |
| PRD | Permanent Reference Document |
| PS | Packet Switched |
| QCI | Quality of Service Class Identifier |
| QoS | Quality of Service |
| RCS | Rich Communication Suite |
| 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 |
| RRAM | RCS Recorded Audio Message |
| RTCP | Real-time Transport Control Protocol |
| RTP | Real-time Transport Protocol |
| SBC | Session Border Controller |
| S-CSCF | Serving Call Session Control Function |
| SDP | Session Description Protocol |
| SDES | Session Description Protocol Security Descriptions for Media Streams |
| 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 |

| Term | Description |
|---|---|
| SIPoTLS | Session Initiation Protocol over Transport Layer Security |
| 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 |
| SPS | Sequence Parameter Set |
| SR | Sender Report |
| SRTP | Secure Real-time Transport Protocol |
| SR-VCC | Single Radio Voice Call Continuity |
| SSO | Single Sign On |
| STUN | Simple Traversal of User Datagram Protocol through Network Address Translations |
| Suggested Chip List | See [PRD-RCC.71] |
| SUPL | Secure User Plane Location |
| 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 |
| UCS2 | 2-byte Universal Character Set |
| UDP | User Datagram Protocol |
| UE | User Equipment |
| UI | User Interface |
| UID | Unique IDentifier |
| 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 |
| VoLTE | Voice over Long Term Evolution |
| VoWiFi | Voice over EPC-integrated Wi-Fi as specified in [PRD-IR.51] |

| Term | Description |
|---|---|
| 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 |
| XML | Extensible Markup Language |
| 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 23.140] | 3GPP TS 23.140 Release 6, 3rd Generation Partnership Project; Multimedia Messaging Service (MMS);Functional description; Stage 2 http://www.3gpp.org |
| **4** | [3GPP TS 23.167] | 3GPP TS 23.167 Release 11, 3rd Generation Partnership Project; IP Multimedia Subsystem (IMS) emergency sessions 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 24.008] | 3GPP TS 24.008 Release 12, 3rd Generation Partnership Project, Mobile radio interface Layer 3 specification; Core network protocols http://www.3gpp.org |
| **7** | [3GPP TS 24.109] | 3GPP TS 24.109 Release 10, 3rd Generation Partnership Project; Bootstrapping interface (Ub) and network application function interface (Ua); Protocol details http://www.3gpp.org |
| **8** | [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 |
| **9** | [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 |
| **10** | [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 |

| Ref | Document Number | Title |
|---|---|---|
| **11** | [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)<br>http://www.3gpp.org |
| **12** | [3GPP TS 24.229-rel15] | 3GPP TS 24.229 Release 15, 3rd Generation Partnership Project; IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP)<br>http://www.3gpp.org |
| **13** | [3GPP TS 24.275] | 3GPP TS 24.275 3rd Generation Partnership Project;<br>Management Object (MO) for Basic Communication Part (BCP) of IMS Multimedia Telephony (MMTEL) communication service<br>http://www.3gpp.org |
| **14** | [3GPP TS 24.279] | 3GPP TS 24.279 Release 8, 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Combining Circuit Switched (CS) and IP Multimedia Subsystem (IMS) services; Stage 3<br>http://www.3gpp.org |
| **15** | [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<br>http://www.3gpp.org |
| **16** | [3GPP TS 24.368] | 3GPP TS 24.368, 3rd Generation Partnership Project; Non-Access Stratum (NAS) configuration Management Object (MO)<br>http://www.3gpp.org |
| **17** | [3GPP TS 26.114] | 3GPP TS 26.114 Release 12, 3rd Generation Partnership Project; IP Multimedia Subsystem (IMS); Multimedia telephony; Media handling and interaction<br>http://www.3gpp.org |
| **18** | [3GPP TS 33.203] | 3GPP TS 33.203 Release 10, 3rd Generation Partnership Project; 3G security; Access security for IP-based services<br>http://www.3gpp.org |
| **19** | [3GPP TS 33.220] | 3GPP TS 33.220 Release 10, 3rd Generation Partnership Project; Generic Authentication Architecture (GAA);Generic Bootstrapping Architecture (GBA)<br>http://www.3gpp.org |
| **20** | [3GPP TS 33.328] | 3GPP TS 33.328 Release 10, 3rd Generation Partnership Project; IP Multimedia Subsystem (IMS) media plane security<br>http://www.3gpp.org |
| **21** | [IETF-DRAFT-RKEEP] | Indication of support for reverse keep-alive, Version 00, June 21, 2012,<br>http://tools.ietf.org/html/draft-holmberg-sipcore-rkeep-00 |
| **22** | [PRD-IR.51] | GSMA PRD IR.51 - “IMS Profile for Voice, Video and SMS over untrusted Wi-Fi access” Version 6.0, 01 May 2018<br>http://www.gsma.com/ |

| Ref | Document Number | Title |
|---|---|---|
| **23** | [PRD-IR.61] | GSMA PRD IR.61 - “Wi-Fi Roaming Guidelines” Version 12.0, 27 September 2017<br>http://www.gsma.com/ |
| **24** | [PRD-IR.65] | GSMA PRD IR.65 - “IMS Roaming, Interconnection and Interworking Guidelines” Version 28.0, 02 May 2018<br>http://www.gsma.com/ |
| **25** | [PRD-IR.67] | GSMA PRD IR.67 - “DNS and ENUM Guidelines for Service Providers and GRX and IPX Providers” Version 15.0, 29 June 2018<br>http://www.gsma.com/ |
| **26** | [PRD-IR.88] | GSMA PRD IR.88 - “LTE and EPC Roaming Guidelines” 18.0, 07 June 2018<br>http://www.gsma.com/ |
| **27** | [PRD-IR.90] | GSMA PRD IR.90 - “RCS Interworking Guidelines” Version 14.0, 13 October 2017<br>http://www.gsma.com/ |
| **28** | [PRD-IR.92] | GSMA PRD IR.92 - “IMS Profile for Voice and SMS” Version 12.0, 02 May 2018<br>http://www.gsma.com/ |
| **29** | [PRD-IR.94] | GSMA PRD IR.94 - “IMS Profile for Conversational Video Service” Version 13.0, 21 June 2018<br>http://www.gsma.com/ |
| **30** | [PRD-NG.102] | GSMA PRD NG.102 - “IMS Profile for Converged IP Communications” Version 5.0, 29 June 2018<br>http://www.gsma.com/ |
| **31** | [RCS-CPM-CONVFUNC-ENDORS] | GSMA PRD RCC.11 RCS Endorsement of OMA CPM 2.2 Conversation Functions, Version 8.0, 06 December 2018<br>http://www.gsma.com/ |
| **32** | [RCS-CPM-IW-ENDORS] | GSMA PRD RCC.10 RCS Endorsement of OMA CPM 2.2 Interworking, Version 8.0, 06 December 2018<br>http://www.gsma.com/ |
| **33** | [RCS-3GPP-SMSIW-ENDORS] | GSMA PRD RCC.08 RCS Endorsement of 3GPP TS 29.311 Service level Interworking for Messaging Services, Version 8.0, 06 December 2018<br>http://www.gsma.com/ |
| **34** | [PRD-RCC.07v8.0] | GSMA PRD RCC.07 Rich Communication Suite 7.0 Advanced Communications Services and Client Specification, Version 8.0, 28 June 2017<br>http://www.gsma.com/ |
| **35** | [PRD-RCC.14] | GSMA PRD RCC.14 HTTP-based Service Provider Device Configuration, Version 6.0, 06 December 2018<br>http://www.gsma.com/ |

| Ref | Document Number | Title |
|---|---|---|
| **36** | [PRD-RCC.15] | GSMA PRD RCC.15 IMS Device Configuration and Supporting Services, Version 6.0, 06 December 2018<br>http://www.gsma.com/ |
| **37** | [PRD-RCC.20] | GSMA PRD RCC.20, Enriched Calling Technical Specification, Version 5.0, 06 December 2018<br>http://www.gsma.com |
| **38** | [PRD-RCC.71] | GSMA PRD RCC.71 RCS Universal Profile Service Description Document, Version 2.3, 06 December 2018<br>http://www.gsma.com |
| **39** | [PRD-RCC.53] | RCS Device API 1.5.1 Specification, Version 3.0, 23 June 2016<br>http://www.gsma.com/ |
| **40** | [RFC2045] | MIME (Multipurpose Internet Mail Extensions) Part One: Format of Internet Message Bodies Text IETF RFC<br>http://tools.ietf.org/html/rfc2045 |
| **41** | [RFC2047] | MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text IETF RFC<br>http://tools.ietf.org/html/rfc2047 |
| **42** | [RFC2396] | Uniform Resource Identifiers (URI): Generic Syntax IETF RFC<br>http://tools.ietf.org/html/rfc2396 |
| **43** | [RFC2425] | A MIME Content-Type for Directory Information IETF RFC<br>http://tools.ietf.org/html/rfc2425 |
| **44** | [RFC2426] | vCard MIME Directory Profile IETF RFC<br>http://tools.ietf.org/html/rfc2426 |
| **45** | [RFC2716] | HTTP Authentication: Basic and Digest Access Authentication IETF RFC<br>http://tools.ietf.org/html/rfc2716 |
| **46** | [RFC3261] | SIP (Session Initiation Protocol) IETF RFC<br>http://tools.ietf.org/html/rfc3261 |
| **47** | [RFC3263] | Session Initiation Protocol (SIP): Locating SIP Servers IETF RFC<br>http://tools.ietf.org/html/rfc3263 |
| **48** | [RFC3264] | An Offer/Answer Model Session Description Protocol IETF RFC<br>http://tools.ietf.org/html/rfc3264 |
| **49** | [RFC3323] | A Privacy Mechanism for the Session Initiation Protocol (SIP) IETF RFC<br>http://tools.ietf.org/html/rfc3323 |
| **50** | [RFC3326] | The Reason Header Field for the Session Initiation Protocol (SIP) IETF RFC<br>http://tools.ietf.org/html/rfc3326 |
| **51** | [RFC3329] | Security Mechanism Agreement for the Session Initiation Protocol (SIP) IETF RFC<br>http://tools.ietf.org/html/rfc3329 |

| Ref | Document Number | Title |
|---|---|---|
| **52** | [RFC3458] | Message Context for Internet Mail IETF RFC<br>http://tools.ietf.org/html/rfc3458 |
| **53** | [RFC3501] | Internet Message Access Protocol - Version 4rev1, IETF RFC<br>http://tools.ietf.org/html/rfc3501 |
| **54** | [RFC3711] | The Secure Real-time Transport Protocol (SRTP), IETF RFC<br>http://tools.ietf.org/html/rfc3711 |
| **55** | [RFC3840] | Indicating User Agent Capabilities in the Session Initiation Protocol (SIP), IETF RFC<br>http://tools.ietf.org/html/rfc3840 |
| **56** | [RFC3862] | Common Presence and Instant Messaging (CPIM): Message Format, IETF RFC<br>http://tools.ietf.org/html/rfc3862 |
| **57** | [RFC3863] | Presence Information Data Format (PIDF), IETF RFC<br>http://tools.ietf.org/html/rfc3863 |
| **58** | [RFC3903] | Session Initiation Protocol (SIP) Extension for Event State Publication IETF RFC<br>http://tools.ietf.org/html/rfc3903 |
| **59** | [RFC3966] | The tel URI for Telephone Numbers, IETF RFC<br>http://tools.ietf.org/html/rfc3966 |
| **60** | [RFC3986] | Uniform Resource Identifier (URI): Generic Syntax, IETF RFC<br>http://tools.ietf.org/html/rfc3986 |
| **61** | [RFC4028] | The Session Timers in the Session Initiation Protocol (SIP), IETF RFC<br>http://tools.ietf.org/html/rfc4028 |
| **62** | [RFC4122] | The Universally Unique IDentifier (UUID) URN Namespace, IETF RFC<br>http://tools.ietf.org/html/rfc4122 |
| **63** | [RFC4479] | A Data Model for Presence, IETF RFC<br>http://tools.ietf.org/html/rfc4479 |
| **64** | [RFC4480] | RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF), IETF RFC<br>http://tools.ietf.org/html/rfc4480 |
| **65** | [RFC4568] | Session Description Protocol (SDP) Security Descriptions for Media Streams, IETF RFC<br>http://tools.ietf.org/html/rfc4568 |
| **66** | [RFC4572] | Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP), IETF RFC<br>http://tools.ietf.org/html/rfc4572 |
| **67** | [RFC4575] | A Session Initiation Protocol (SIP) Event Package for Conference State, IETF RFC<br>http://tools.ietf.org/html/rfc4575 |

| Ref | Document Number | Title |
|---|---|---|
| **68** | [RFC4826] | Extensible Markup Language (XML) Formats for Representing Resource Lists, IETF RFC<br>http://tools.ietf.org/html/rfc4826 |
| **69** | [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<br>http://tools.ietf.org/html/rfc4867 |
| **70** | [RFC4961] | Symmetric RTP / RTP Control Protocol (RTCP), IETF RFC<br>http://tools.ietf.org/html/rfc4961 |
| **71** | [RFC4975] | The Message Session Relay Protocol (MSRP), IETF RFC<br>http://tools.ietf.org/html/rfc4975 |
| **72** | [RFC5104] | Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF), IETF RFC<br>http://tools.ietf.org/html/rfc5104 |
| **73** | [RFC5196] | Session Initiation Protocol (SIP) User Agent Capability Extension to Presence Information Data Format (PIDF), IETF RFC<br>http://tools.ietf.org/html/rfc5196 |
| **74** | [RFC5322] | Internet Message Format IETF RFC<br>http://tools.ietf.org/html/rfc5322 |
| **75** | [RFC5365] | Multiple-Recipient MESSAGE Requests in the Session Initiation Protocol (SIP), IETF RFC<br>http://tools.ietf.org/html/rfc5365 |
| **76** | [RFC5438] | Instant Message Disposition Notification (IMDN), IETF RFC<br>http://tools.ietf.org/html/rfc5438 |
| **77** | [RFC5438Errata] | Instant Message Disposition Notification (IMDN), IETF RFC 5438 Errata ID 3013<br>http://www.rfc-editor.org/errata_search.php?rfc=5438 (see also section C.2) |
| **78** | [RFC5491] | GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations, IETF RFC<br>http://tools.ietf.org/html/rfc5491 |
| **79** | [RFC5547] | A Session Description Protocol (SDP) Offer/Answer Mechanism to Enable File Transfer, IETF RFC<br>http://tools.ietf.org/html/rfc5547 |
| **80** | [RFC5626] | Managing Client-Initiated Connections in the Session Initiation Protocol (SIP), IETF RFC<br>http://tools.ietf.org/html/rfc5626 |
| **81** | [RFC5724] | URI Scheme for Global System for Mobile Communications (GSM) Short Message Service (SMS), IETF RFC<br>http://tools.ietf.org/html/rfc5724 |

| Ref | Document Number | Title |
|---|---|---|
| **82** | [RFC5870] | A Uniform Resource Identifier for Geographic Locations ('geo' URI), IETF RFC<br>http://tools.ietf.org/html/rfc5870 |
| **83** | [RFC6068] | The 'mailto' URI Scheme IETF RFC<br>http://tools.ietf.org/html/rfc6068 |
| **84** | [RFC6135] | Alternative Connection Model for the Message Session Relay Protocol (MSRP), IETF RFC<br>http://tools.ietf.org/html/rfc6135 |
| **85** | [RFC6223] | Indication of Support for Keep-Alive, IETF RFC<br>http://tools.ietf.org/html/rfc6223 |
| **86** | [RFC6265] | HTTP State Management Mechanism, IETF RFC<br>http://tools.ietf.org/html/rfc6265 |
| **87** | [RFC6350] | vCard Format Specification, IETF RFC<br>http://tools.ietf.org/html/rfc6350 |
| **88** | [RFC6665] | SIP-Specific Event Notification, IETF RFC<br>http://tools.ietf.org/html/rfc6665 |
| **89** | [RFC7230] | Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing, IETF RFC<br>http://tools.ietf.org/html/rfc7230 |
| **90** | [RFC7232] | Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests IETF RFC<br>http://tools.ietf.org/html/rfc7232 |
| **91** | [RFC7233] | Hypertext Transfer Protocol (HTTP/1.1): Range Requests IETF RFC<br>http://tools.ietf.org/html/rfc7233 |
| **92** | [RFC7235] | Hypertext Transfer Protocol (HTTP/1.1): Authentication IETF RFC<br>http://tools.ietf.org/html/rfc7235 |
| **93** | [RFC7515] | JSON Web Signature (JWS) IETF RFC<br>http://tools.ietf.org/html/rfc7515 |
| **94** | [RFC7517] | JSON Web Key (JWK) IETF RFC<br>http://tools.ietf.org/html/rfc7517 |
| **95** | [RFC7518] | JSON Web Algorithms (JWA) IETF RFC<br>http://tools.ietf.org/html/rfc7518 |
| **96** | [RFC7638] | JSON Web Key (JWK) Thumbprint IETF RFC<br>https://tools.ietf.org/html/rfc7638 |
| **97** | [GML3.1.1] | OpenGIS® Geography Markup Language (GML) Implementation Specification, Version 3.1.1, OGC 03-105r1<br>http://www.opengeospatial.org/ |
| **98** | [CAB_TS] | OMA Converged Address Book (CAB) Specification, Approved Version 1.0, 13 November 2012<br>http://www.openmobilealliance.org |

| Ref | Document Number | Title |
|---|---|---|
| **99** | [CPM-SYS_DESC] | OMA Converged IP Messaging System Description, Candidate Version 1.0, 12 October 2010<br>http://www.openmobilealliance.org |
| **100** | [CPM-MSGSTOR-REST] | CPM Message Store using RestFul API, Draft Version 1.0, 16 May 2017, OMA-TS-CPM_Message_Store_Using_RESTFul_API-V1_0-20170516-D<br>http://www.openmobilealliance.org |
| **101** | [MMSCONF] | OMA MMS Conformance Document, Approved Version 1.3, 13 September 2011<br>http://www.openmobilealliance.org |
| **102** | [MMSCTR] | OMA Multimedia Messaging Service Client Transactions, Approved Version 1.3, 13 September 2011<br>http://www.openmobilealliance.org |
| **103** | [MMSENC] | OMA Multimedia Messaging Service – Encapsulation Protocol, Approved Version 1.3 – 13 Sep 2011<br>http://www.openmobilealliance.org |
| **104** | [Presence] | OMA Presence SIMPLE Specification, 1.1,<br>http://www.openmobilealliance.org/ |
| **105** | [Presence2.0_DDS] | Presence SIMPLE Data Specification, Approved Version 2.0, 29 September 2009<br>http://www.openmobilealliance.org/ |
| **106** | [Presence2.0_TS] | Presence SIMPLE Specification, Approved Version 2.0, 10 July 2012<br>http://www.openmobilealliance.org/ |
| **107** | [PDE_14] | OMA Presence SIMPLE Data Extensions, Approved Version 1.4, 22 December 2015<br>http://www.openmobilealliance.org/ |
| **108** | [PRESENCE2MO] | OMA Management Object for Presence SIMPLE 2.0, Approved Version 2.0, 10 July 2012<br>http://www.openmobilealliance.org |
| **109** | [vCard21] | vCard, The Electronic Business Card, A versit Consortium Specification, 18 September 1996<br>http://www.imc.org/pdi/vcard-21.doc |
| **110** | [ISO 639-1] | ISO 639-1:2002 Codes for the representation of names of languages -- Part 1: Alpha-2 code, July 2002<br>http://www.iso.org |
| **111** | [ISO8601] | ISO 8601:2004 Data elements and interchange formats -- Information interchange -- Representation of dates and times, 18 March 2008<br>http://www.iso.org |
| **112** | [POSIX] | IEEE Standard for Information Technology—Portable Operating System Interface<br>IEEE Std 1003.1, 2013 Edition |

| Ref | Document Number | Title |
|---|---|---|
| **113** | [HTML-4.0] | HTML 4.01 Specification<br>https://www.w3.org/ |
| **114** | [OpenID Connect] | OpenID Connect Core; OpenID Foundation<br>http://openid.net/connect/ |

## 1.5 Differences to previous specifications

RCS 9.0 documented in the current version of RCC.07 evolves on the functionality defined for RCS 8.0 (as defined in RCC.07 v9.0).

The following sub-sections list the major differences.

### 1.5.1 New features and procedures

- Configuration
  - Support for EAP AKA as a means to authenticate a configuration request (see [PRD-RCC.14])
- Messaging
  - Support for large Pager Mode Standalone Messages (sections 2.4.4.1, 3.2.2.7, A.1.3 and A.2.4)
  - Support for revocation of Standalone Messages (section 3.2.2.5.2)
  - Provide a means to indicate capabilities in a Pager Mode Standalone Message (section 3.2.2.6)
  - Chatbots
    - Enable use of Standalone Messaging for Chatbots (sections 2.4.4.1, 2.6.1.3, 3.2.2.5.1, 3.6.2.1, 3.6.8, 3.6.9 and A.1.3) including selection logic for the appropriate technology (section 3.2.1.2)
    - Procedure to fall back to Chatbot Communication when using P2P Messaging to contact a Chatbot (in case capability exchange has been disabled, section 3.2.1.1, 3.6.2.4, 3.6.8)
- Enriched Calling
  - Provide a procedure to use the SIP headers in end-to-end VoLTE calls to provide the pre-call service (see [PRD-RCC.20] and sections 2.4.4.1, 2.6.1.3 and 3.3.2)

### 1.5.2 Removed features and procedures

Removed NO MSRP SUPPORT and CALL COMPOSER TIMER IDLE client configuration parameters (see sections 2.8.1.4, 2.8.1.6, A.1.5, A.1.6.2, A.1.10, A.2 and [PRD-RCC.20]).

Removed Video Share (See section 2.6.1.3, 3.3.1, A.1.5, A.1.14.1, A.2.2 and [PRD-RCC.20])

Removed client procedure for automatic rejoin of an active Group Chat (see [RCS-CPM-CONVFUNC-ENDORS] and section B.1)

### 1.5.3 Modified features and procedures

| Area | Section | Differences from RCS 8.0 |
|---|---|---|
| Configuration | 2.3.2.2, 2.4.4.1, 2.8.1.5, 3.2.8.3.2, 3.6.3.3, A.1.14, A.2.2, A.3, and [PRD-RCC.14] | Align Data Off configuration with 3GPP procedures |
| Configuration | [PRD-RCC.14] | Provide support for GSM 7 bit alphabet for sending an SMS with a OTP value |
| IMS Registration and signalling | C.4.1 | Clarified format of the User-Agent and Server header |
| Security | 2.12.1.1.2, 3.2.5.3.1.1 and 4.1.4 | Changed RFC referred to for SIP and HTTP Digest to [RFC2716] and mandated use of qop directive where appropriate |
| Messaging | 3.2.3.3, B.1 | Support procedure with specific session for deferred delivery of IMDNs related to a Chat Message only on client for backward compatibility with earlier versions. |
| Messaging | 3.2.5.3.2.1, 4.1.8, 4.1.12.1.4, 4.1.13.1, 4.1.16.8 | Clarified that the direction in which a message was sent is indicated through the Direction attribute on the Message Store interface (rather than Message-Direction) |
| Messaging | 4.1.12.1.4, 4.1.12.2, 4.1.13.1, 4.1.13.2 | Clarified the use of the CPIM attribute when storing xMS messages in the Message Store |
| Messaging | [RCS-CPM-CONVFUNC-ENDORS] | Aligned with updated CPM specification containing several bugfixes |
| File Transfer | 4.1.15 | Allow download to the terminating network of localised File Transfers according to Service Provider Policy |
| Chatbots | 3.6.5.1.4.1 | Clarified that Chatbot Platform AF needs to include a 'tk' URI parameter in the SIP responses that it sends |
| Chatbots | 3.6.8.4, 3.6.8.9, A.1.3 and A.2.4 | Clarified the client authorisation to use the Chatbot Service |
| Chatbots | 3.6.9 | Clarified that Deferred Messaging also applies to disposition notifications |
| Chatbots | 3.6.10.6.2 | Clarified the procedures for sending payload from client to Chatbot Platform in reaction to a suggested reply or action (including the requestDeviceSpecifics action) |

**Table 1: Modifications from RCS 8.0 that was defined in RCC.07 version 9.0**

# 2 RCS General Procedures

## 2.1 RCS architecture

For RCS, the base 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 1 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 Network-to-Network Interface (NNI) aspects.

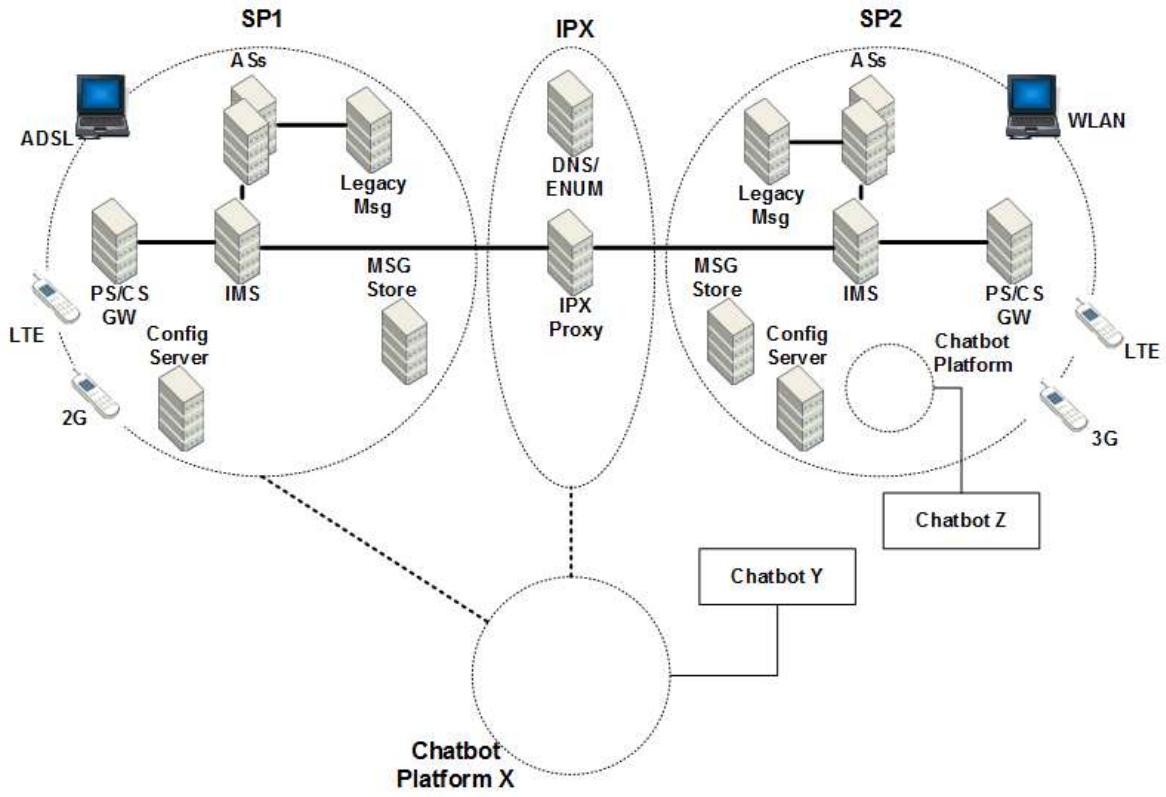



**Figure 1: Simplified Example of RCS Architecture**

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). MSG Store relates to the CPM (Converged IP Messaging) Message Store Server as illustrated in section 4.1. 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
- Multimedia Telephony (MMTEL) Application Server
- ASs for the support of Chatbot Functionality (see section 3.6)

An Autoconfiguration Server is used to provide the clients with the configuration to support RCS services.

Figure 1 shows examples of two RCS Service Providers exchanging traffic with each other using the standard 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 1. 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 2.6.1.3. It is recommended to follow the IMS node naming guidelines as defined in [PRD-IR.67] for naming the Messaging Server, since this server is required to be addressable across IPX.

RCS also provides support for Chatbot communications (see [PRD-RCC.71] for a definition and use cases) through the integration of Chatbot Platforms (see [PRD-RCC.71] for a definition) in the overall architecture. These platforms can either connect through the interconnect infrastructure or connect directly to an RCS Service Provider's network. Further details on Chatbot Platforms and their integration into the RCS architecture are provided in section 3.6 and section 3.6.1 in particular.

## 2.2 RCS devices and client types

RCS defines two types of devices:

1. **Primary device**: a device carrying a Subscriber Identity Module (SIM) that is associated with the identity (i.e. IMPU/MSISDN) used for RCS. Two types of RCS clients exist for such a device that connect directly to the IMS:

    a) **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.). This type of RCS client shall represent the identity of the device as per [PRD-NG.102] when enabled for VoLTE and Voice over Wi-Fi (VoWiFi). Otherwise, section 2.4.2 and [3GPP TS 24.229] apply and the International Mobile Station Equipment Identity (IMEI) shall be used in sip.instance during registration.

    b) **RCS downloadable client**: This is a client providing its own IMS connectivity 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.

NOTE1: Next to this, there may also be downloadable clients that use terminal APIs (e.g. [PRD-RCC.53]) to access the RCS functionality that is provided by a device's RCS embedded client. This type of client is not considered in this document because it does not alter the UNI which is handled by the device's RCS embedded client.

2. **Secondary device**: a device that does not carry a SIM that is associated to the identity used for RCS

NOTE2: It may happen that a secondary device carries a SIM (e.g. a tablet or PC providing cellular data connectivity). That SIM will be associated to a different identity than the one used for RCS though.

RCS services can also be deployed using an identity that is not linked to a mobile network (e.g. a fixed-line telephone number or other identity). [PRD-RCC.14] describes in section 2.10 a generic mechanism for the configuration of such clients. Because it will be dependent

on the use case, it is out of scope of this document how an identity is assigned to such devices and on which basis they are considered as a primary or secondary device.

## 2.3 Configuration Procedures

### 2.3.1 Client configuration parameters

The client shall offer individual RCS services to the user only if the Service Provider has authorised the use via the relevant client configuration parameters. In addition, the Service Provider need to provide IMS Core network configuration data to the client. RCS clients shall support the procedures for client configuration described in this section.

If the client is not configured with service configuration data for RCS or if RCS service are disabled via configuration, then the client shall not offer the relevant RCS service(s) to the user and shall disable the service specific entry points. Exceptions apply for IP Voice Call and IP Video Call Services as described in section 3.4 and 3.5, if configuration data is provided to the device via other sources.

The set of client configuration parameters relevant for RCS services is defined in Annex A Managed objects and configuration parameters.

Devices enabled for VoLTE or VoWiFi and conversational video are configured for IP Voice Call and IP Video via the parameters defined in [PRD-IR.92], [PRD-IR.51] and [PRD-IR.94] in addition to the parameters defined in section A.1.12.

Some IMS Core network related configuration parameters are not applicable if the device registers services via the IMS well-known APN as defined in [PRD-RCC.15].

NOTE: Secondary devices (see section 2.2) cannot be enabled for VoLTE or VoWiFi. Similarly, downloadable clients (see section 2.2) do not have sufficient device integration to behave as a device that is enabled for VoLTE or VoWiFi.

If the configuration data resulting from the client provisioning authorises the client to use RCS services, then the RCS client shall register with the network in accordance with definitions for the services described in this document. Once this registration process has successfully completed, the user is able to make use of the RCS services.

Client configuration parameters could also be updated and withdrawn by the Service Provider using the mechanism described in this section.

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

### 2.3.2 RCS client autoconfiguration mechanisms

#### 2.3.2.1 Overview

For the management of the configuration parameters controlling the RCS functionality in devices carrying the SIM associated with an RCS user’s main identity, the mechanism defined in [PRD-RCC.14] including the enhancements described in section 2.1 of [PRD-RCC.15] shall be used.

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 [PRD-RCC.14] and [PRD-RCC.15] for such devices.

RCS devices supporting HTTP configuration as described in [PRD-RCC.14] shall support all enhancements described in section 2.1 of [PRD-RCC.15] including those relying on End User Configuration Requests (EUCR) covered in sections 2.1.2 and 2.1.3.1 of [PRD-RCC.15].

#### 2.3.2.2 Configuration HTTP GET request parameters

The RCS client shall include in the configuration HTTP GET request the generic request parameters as defined in [PRD-RCC.14].

The RCS client shall indicate the support of RCS services by inclusion of an "app" HTTP GET request parameter as defined in [PRD-RCC.14] with the value set to "ap2002".

The RCS client shall include in the configuration HTTP GET request the parameters defined in [PRD-RCC.15].

If supported, the RCS client shall indicate the support of the Non-Access Stratum Management Object defined [3GPP TS 24.368] by inclusion of an "app" HTTP GET request configuration parameter as defined in [PRD-RCC.14] with the value set to "urn:oma:mo:ext-3gpp-nas-config:1.0".

The RCS client shall also include in the configuration HTTP GET request the parameters defined in Table 2:

| Parameter | Description | Mandatory | Format |
|---|---|---|---|
| rcs_state | This is either -4, -3, -2, -1, 0 or a positive integer.<br>The parameter value shall be set to:<br>0 indicating that no configuration exists (e.g. out of the box, after a SIM change), or if the client received in the previous configuration response a version parameter of the VERS characteristic set to "0" or a negative integer value, or if the client received in the previous configuration response a RCS DISABLED STATE parameter set to "0"..<br>A positive value indicates the version of the configuration document in which the last RCS configuration was received i.e. in most cases this will match the value of the generic *vers* parameter defined in [PRD-RCC.14].<br>-1 indicating that the client has received in the last configuration document the RCS DISABLED STATE configuration parameter set to -1 (i.e. the operator has disabled the RCS services on the device/client).<br>-2 indicating that the last configuration | Y | Int (-4,-3, -2, -1, 0 or a positive integer) |

| Parameter | Description | Mandatory | Format |
|---|---|---|---|
| | document included a RCS DISABLED STATE configuration parameter set to -2 (i.e. RCS is disabled on the device, but a configuration query might be triggered on user action).<br>-3 indicating that the last configuration document included a RCS DISABLED STATE configuration parameter set to -3 (i.e. RCS is in a dormant state (i.e. no registration) in a way that is transparent to the user).<br>-4 indicating that the user has explicitly disabled the RCS services on the device/client.<br>A positive value indicates the version of the configuration document in which the last RCS configuration was received i.e. in most cases this will match the value of the generic *vers* parameter defined in [PRD-RCC.14]. | | |
| rcs_version | String that identifies the RCS version supported by the client.<br>For this release it shall be set to “9.0” (without the quotes) | Y | 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).<br>In case multiple, (potentially overlapping) sets are supported the parameter shall be included multiple times | N | String (15 max), Case-Sensitive<br>Multi-valued |
| 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.<br><br>client_version_value = Platform "-" VersionMajor "." VersionMinor<br>Platform = Alphanumeric (9 max)<br>VersionMajor = Number (2 char max)<br>VersionMinor = Number (2 char max)<br><br>Example:<br>client_version=RCSAndrd-1.0 | Y | String<br>(15 max), Case-Sensitive |

| Parameter | Description | Mandatory | Format |
|---|---|---|---|
| default_sms_app | This is either 0,1 or 2<br>0 indicates that the OS does not allow user to select SMS application or the client cannot identify the selected SMS application<br>1 indicates that the RCS messaging client is selected as the default SMS application<br>2 indicates that the RCS messaging client is not selected as the default SMS application | N, only mandatory for client platforms supporting SMS (i.e. primary devices) | Int (0,1,2) |

**Table 2: HTTP configuration: additional RCS specific HTTPS request GET parameters**

#### 2.3.2.3 Configuration request triggers

In addition to the triggers for client configuration defined in [PRD-RCC.14] and [PRD-RCC.15], the following additional triggers apply for RCS:

- if the RCS client is enabled (i.e. the last received configuration XML document
  - did include a positive integer value in the version parameter of the VERS characteristic defined in [PRD-RCC.14], and
  - did not include the RCS DISABLED STATE configuration parameter, see section A.1.1),

  and the user changes the selected SMS application resulting in a different value of the default_sms_app parameter, this shall also trigger a configuration query.

- If the user disables all RCS services in the client settings, then the client shall send a configuration request with the rcs_state set to “-4”. As a result of the processing of the configuration request, the client shall process any response in accordance with the client configuration response procedures. The client shall respect the user's request to disable RCS services.
- If RCS services are disabled on the client as described above and the user enables all RCS services in the client settings, then the client shall send a configuration request with the rcs_state set to the value stored locally on the client.
- If the IMS registration fails as described in section 2.12.1.1.2, the device shall consider the device configuration invalid and perform a device configuration request as defined in section 2.12.1.1.2.
- If the HTTP Content Server returns a HTTP 401 AUTHENTICATION REQUIRED error response to the client's HTTP POST request using the values of the configuration parameters FT HTTP CS USER and FT HTTP CS PWD for authentication as defined in section 3.2.5.3.1.
- If the HTTP Content Server or the Localisation function returns a HTTP 401 AUTHENTICATION REQUIRED error response to the client's HTTP GET request using the values of the configuration parameters FT HTTP CS USER and FT HTTP CS PWD for authentication as defined in section 3.2.5.3.2.
- If the Common Message Store returns an authentication failure in result of the RESTful API authentication procedure using the values of the configuration parameters defined in Annex A.1.3.

#### 2.3.2.4 RCS version management

The client configuration mechanism supports a mechanism for RCS version management to support changes of the functionality and configuration data version caused by RCS releases and set of RCS services profiles in external documents.

The configuration parameter RCS DISABLED STATE in conjunction with the SUPPORTED RCS VERSION and SUPPORTED RCS PROFILE VERSION can be used for RCS version management as described in 2.3.2.4.

The client shall indicate the RCS release version for which it requests configuration data via the rcs_version request parameter. In addition, it may request configuration data for one or more sets of profiled RCS services via the rcs_profile configuration request parameter.

The configuration server shall be able to indicate to the client the RCS versions it supports via the SUPPORTED RCS VERSIONS configuration parameter as defined in section A.1.1.

The configuration server shall be able to indicate to the client the RCS profile versions it supports via the SUPPORTED RCS PROFILE VERSIONS configuration parameter as defined in section A.1.1.

If a configuration server cannot fully satisfy the client request for a specific RCS version or RCS profile versions, then it may disable RCS services by means of the RCS DISABLED STATE defined in section A1.1. The client shall use the values of the SUPPORTED RCS VERSIONS and SUPPORTED RCS PROFILE VERSIONS configuration parameters to adapt its own behaviour and the client configuration request to enable RCS services via alternative versions.

The client shall inspect the values of the SUPPORTED RCS VERSIONS and SUPPORTED RCS PROFILE VERSIONS configuration parameters for higher than the current negotiated version, if a valid configuration is provided by the configuration server. If a higher version is available and the client supports it, then the client shall change behaviour and the request parameters for a subsequent configuration request to update functionality.

#### 2.3.2.5 RCS client management

During configuration the service provider may want to control the general state of the RCS client (e.g. disable it) without affecting the other services configured through the same configuration document. This shall be done through the generic RCS DISABLED STATE client configuration parameter defined in section A.1.1. Only if not included, other RCS settings shall be included in the configuration document. The client shall ignore any such settings if included when also the RCS DISABLED STATE parameter is included in the document.

If the Service Provider chooses to temporary disable RCS functionality on the device/client, the RCS DISABLED STATE configuration parameter shall be set to 0.

If the Service Provider chooses to permanently disable the RCS functionality on an RCS capable device/client, the RCS DISABLED STATE configuration parameter shall be set to -1.

If the SIM is swapped or the device is reset, the RCS capable device shall again set the rcs_state parameter defined in Table 2 to 0 in the queries for configuration settings.

If the Service Provider chooses to disable the RCS functionality on an RCS capable device/client until there is a User Interface (UI) dependent user action triggering a new query, the RCS DISABLED STATE parameter shall be set to -2.

If the SIM is swapped or the device is reset or the user triggers the UI dependent action, the RCS capable device shall initiate a configuration request including the rcs_state parameter set to 0.

If the Service Provider chooses to put the RCS functionality on an RCS capable device/client in a dormant state, the RCS DISABLED STATE parameter shall be set to -3.

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 rcs_state parameter '-3'.
- The existing configuration document remains valid (i.e. a response with the RCS DISABLED STATE parameter set to '-3' shall be handled in this aspect as if the rcs_state 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 that does not contain the RCS DISABLED STATE configuration parameter. If it was registered when the document with the RCS DISABLED STATE configuration parameter set to '-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) including the rcs_state parameter with as value the version of the last configuration document that included a full RCS configuration (i.e. not including the RCS DISABLED STATE parameter).
    - 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.

## 2.4 IMS registration

### 2.4.1 General

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).

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

As specified in [3GPP TS 24.229], the SIP REGISTER request shall be sent to the IP address and port obtained via the discovery procedure (see section 2.4.5). If the device was unable to obtain a specific port, then the default port as specified in [RFC3261] shall be used.

The client shall send subsequent SIP REGISTER and non-REGISTER requests to the IP address and port that is used for the initial REGISTER, unless the security mechanism requires the use of negotiated ports for the exchange of protected messages.

In all cases, the device shall register in IMS indicating whether the device can receive SMSs associated with the identity used for RCS when the device is not registered in IMS for messaging by using the telephony feature tag described in section 2.4.3.

The device must use the authentication mechanisms as described in section 2.12.

A precondition to register is that all of the mandatory parameters presented in section A.1 shall be correctly configured. This includes those optional parameters that, due to their dependency on the configured value of a mandatory parameter, have become mandatory.

#### 2.4.1.1 Devices Enabled for VoLTE or VoWiFi

If the device is enabled for VoLTE or VoWiFi then it must additionally follow the procedures for registration specified in [PRD-NG.102]. If the device is configured to not share a registration between Multimedia Telephony (including SMSoIP) and other RCS services (see the configuration parameter RCS VOLTE SINGLE REGISTRATION defined in section A.1.6), a User-Agent header field shall be included in the SIP REGISTER request for the registration for RCS services. This header field shall be formatted as specified in Annex C.4.1. If the device is configured to share a registration for all RCS services (i.e. including Multimedia Telephony and SMSoIP), then the handling of the User-Agent and Server header is out of the scope of this document.

##### 2.4.1.1.1 Use of non-cellular access

When the domain selection has selected IMS voice and the device is configured to share a registration for all RCS services (i.e. including Multimedia Telephony and SMSoIP, see the configuration parameter RCS VOLTE SINGLE REGISTRATION defined in section A.1.6), the device is using VoLTE or VoWiFi, and shall not directly register in non-cellular networks (i.e. it shall not directly register over a Wi-Fi network). The client may register over EPC integrated Wi-Fi according to [PRD-NG.102].

When the domain selection has not selected IMS voice or the device is not configured to share a registration for all RCS services (i.e. including Multimedia Telephony and SMSoIP), the device may register using non-cellular access as described in section 2.4.1.2.1.

As soon as the domain selection is again using IMS voice and a device is configured to share a registration for all RCS services (i.e. including Multimedia Telephony and SMSoIP, see the configuration parameter RCS VOLTE SINGLE REGISTRATION defined in section A.1.6), the device shall attempt to de-register from IMS through the non-cellular access and shall register again using IMS over the cellular or EPC-integrated Wi-Fi network access.

#### 2.4.1.2 Devices not enabled for VoLTE or VoWiFi

If the device is not enabled for VoLTE or VoWiFi as defined in section 2.8.1) and it is configured to support RCS IP Voice Call and/or RCS IP Video Call, it shall always register in IMS taking into account the rules defined in respectively Table 43 and Table 44 in sections 3.4 and 3.5.

NOTE: Secondary devices (see section 2.2) cannot be enabled for VoLTE or VoWiFi. Similarly, downloadable clients (see section 2.2) do not have sufficient device integration to behave as a device that is enabled for VoLTE or VoWiFi and will, therefore, have to assume that the device is not enabled for VoLTE or VoWiFi.

The client shall send a SIP REGISTER message to the network using the configuration parameters (SIP proxy and other IMS parameters as presented in section 2.2.1 of [PRD-RCC.15]). A User-Agent header field shall be included in this SIP REGISTER request. This header field shall be formatted as specified in Annex C.4.1.

##### 2.4.1.2.1 Use of non-cellular access

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

When registered over direct non-cellular access, all RCS traffic including the traffic from supporting protocols (i.e. XCAP and HTTP) shall use this direct non-cellular connection.

### 2.4.2 Procedures for multidevice handling

A RCS client shall support the Instance ID to allow Application Servers to uniquely address clients residing on different devices as specified in [3GPP TS 24.229] based on the definitions below.

The client shall provide its instance ID in the Contact header field of REGISTER and non-REGISTER requests via the "*sip.instance*" feature tag as described in [3GPP TS 24.229].

If the device is enabled for VoLTE or VoWiFi and other RCS services, then the device shall set the value of sip.instance feature tag as specified in [PRD-NG.102].

If the device is enabled for VoLTE or VoWiFi, then a downloadable RCS client residing on the device shall set the value of sip.instance feature tag to a UUID (Universal Unique Identifier) value.

If the device is not enabled for VoLTE or VoWiFi, then the client shall set the value of the sip.instance feature tag

- to the IMEI value as per [3GPP TS 24.229], if the client has access to the device IMEI, otherwise
- to a UUID value, if the client has no access to the device IMEI or no IMEI is available for the device.

If the client must provide as result of the procedures above a UUID value in the sip.instance feature tag, then the client shall use shall use:

- the UUID value provided via the configuration parameter uuid_Value, as defined in [PRD-RCC.15]) if present in the IMS MO associated with RCS.
- otherwise, if the uuid_Value is absent in the IMS MO associated with RCS, the client shall generate a UUID value as defined in [RFC4122] section 4.2 at the time of the client instantiation, which must not be modified over the lifetime of the client instance.

### 2.4.3 Telephony feature tag

RCS defines a telephony feature tag used to indicate to the IMS network whether the device supports CS telephony services and hence can receive SMSs associated with the identity used for RCS when the device is not registered in IMS for messaging. The feature tag shall be included in the Contact header at registration with possible values to include "*none*" or "*cs "*. The use of any other value carried in the feature tag is out of scope for this specification. If no value is included in the feature tag, it is treated by the IMS network as if it carried the value of "none".

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

Using the terms primary and secondary devices as defined in section 2.2:

- For a secondary device that by definition does not support CS telephony and thus does not support receiving SMSs for the identity used for RCS, the feature tag shall either not be present at all or be set as follows: +g.gsma.rcs.telephony="none".
- For a primary device that supports CS telephony and thus supports receiving SMSs for the identity used for RCS, the feature tag shall be set as follows even when the device is not currently in coverage conditions where CS telephony can be used: +g.gsma.rcs.telephony="cs".

### 2.4.4 Services feature tags

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

Based on the relevant service specifications, the client shall include the following feature tags in the SIP REGISTER request when the corresponding service has been the authorised/enabled for the used access (i.e. cellular or Wi-Fi):

| RCS service | Tags |
|---|---|
| Standalone Messaging (section 3.2.2) | +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.oma.cpm.msg,urn%3Aurn-7%3A3gpp-service.ims.icsi.oma.cpm.largemsg,urn%3Aurn-7%3A3gpp-service.ims.icsi.oma.cpm.deferred";+g.gsma.rcs.cpm.pager-large |
| Chat (section 3.2.3); Group Chat (section 3.2.4) | +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.oma.cpm.session" |
| File Transfer (section 3.2.5) | +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.fthttp"<br>NOTE: This feature tag is included in the registration to support backward compatibility and to inform network elements of the support of the feature. |
| File Transfer via SMS (section 3.2.5.7) | +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.ftsms" |
| Call Composer via Enriched Calling session (section 3.3.2.1) | +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.gsma.callcomposer" |
| Call Composer via Multimedia Telephony session (section 3.3.2.1) | +g.gsma.callcomposer |
| Post Call (section 3.3.2.2) | +g.3gpp.icsi-ref=”urn%3Aurn-7%3A3gpp-service.ims.icsi.gsma.callunanswered” |
| Shared Map (section 3.3.1.2) | +g.3gpp.icsi-ref=”urn%3Aurn-7%3A3gpp-service.ims.icsi.gsma.sharedmap” |
| Shared Sketch (section 3.3.1.3) | +g.3gpp.icsi-ref=”urn%3Aurn-7%3A3gpp-service.ims.icsi.gsma.sharedsketch” |
| RCS IP Voice Call (section 3.4) | +g.gsma.rcs.ipcall; +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel" |
| RCS IP Video Call (section 3.5) | +g.gsma.rcs.ipcall; +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel";video |
| Geolocation PUSH (section 3.2.6) | +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.oma.cpm.filetransfer";+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.geopush"<br>NOTE: These feature tags are included in the registration to support backward compatibility and to inform network elements of the support of the feature. |
| Geolocation PUSH via SMS (section 3.2.6.3) | +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.geosms" |
| Chatbot Communication using sessions (section 3.6.2.1) | +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.chatbot" |
| Chatbot Communication using Standalone Message (section 3.6.2.1) | +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.chatbot.sa" |
| Chatbot version supported (section 3.6.2.2) | +g.gsma.rcs.botversion="#=1" |

| | |
|---|---|
| Plug-ins support (section 3.2.8.2.1) | +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.plugin" |
| CPIM header extension support (section 2.14) | +g.gsma.rcs.cpimext |
| Data Off (see section 2.8.1.5) | +g.3gpp.ps-data-off="active" or<br>+g.3gpp.ps-data-off="inactive" |

**Table 3: RCS Services feature tags**

NOTE: For a device enabled for VoLTE or VoWiFi, the feature tags for the MMTEL and SMS over IP services may be included in addition, as per the procedures in [PRD-NG.102].

If several IMS Application Reference Identifier (IARI) tag values or several IMS Communication Service Identifier (ICSI) tag values are included in a SIP REGISTER request, consistently with [RFC3840] and [3GPP TS 24.229], IARI tag values or ICSI tag values shall be concatenated using commas as shown in the example below:

| |
|---|
| +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.fthttp, urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.geopush" |

**Table 4: IARI tag concatenation format example**

If the same ICSI or IARI tag value or feature tag is used by several services that are enabled, it shall be included only once in the SIP REGISTER request.

### 2.4.5 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 enabled for VoLTE or VoWiFi shall select the P-CSCF as defined in [PRD-NG.102].
- Other devices 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 IMS device management object (as defined in section 2.2.1 of [PRD-RCC.15]) into account if SIP Digest is to be used for authentication, i.e. SIPoUDP (SIP over UDP), SIPoTLS (SIP over Transport Layer Security) or SIPoTCP (SIP over TCP) depending on the access network type (PS or Wi-Fi).
  - Ignore the SIP transport protocol settings in the IMS device management object (as defined in section 2.2.1 of [PRD-RCC.15]) 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 Domain Name System Service [DNS SRV] records) to support load distribution in the network.

### 2.4.6 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.

#### 2.4.6.1 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.

#### 2.4.6.2 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.

#### 2.4.6.3 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.

#### 2.4.6.4 Loss of Connectivity

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

For a client that is not enabled for VoLTE or VoWiFi or a client enabled for VoLTE or VoWiFi that is not configured to share the registration for all RCS services (i.e. including Multimedia Telephony and SMSoIP), 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 2.2.1 of [PRD-RCC.15] for the new access network type) is the same as for the lost connection and the IMS registration is not yet expired, the client 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.

#### 2.4.6.5 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 2.4.6.3.
- 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 2.4.6.4.
- 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.

### 2.4.7 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 2.4.1. 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-Mobile Subscriber Integrated Services Digital Network Number (MSISDN) based identity. The RCS client would in that case be provisioned with only the appropriate SIP URI parameter as seen in section 2.2.1.1.2 of [PRD-RCC.15], 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.51]. 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 2.5.3.4.

NOTE2: The identification in Common Profile for Instant Messaging (CPIM) headers is discussed in section 3.2.3 and 3.2.4.

### 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. The Service Provider shall always provide the *P-Asserted-Identity* header field(s) towards the RCS client. 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 3.2.3.3), 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.

If a matched contact is found for which the telephone number stored in the address book is not international format, the client shall cache for that contact the MSISDN extracted from the signalling for a duration equal to the value configured for the CAPABILITY INFO EXPIRY client configuration parameter defined in section A.1.9.

#### 2.5.2.2 In-call SIP requests

##### 2.5.2.2.1 Caller: relating the ongoing call with in-call incoming SIP requests

The destination identity of the telephony call that the caller dials or gets from his address book and the originator identity of any in-call incoming request may be in various formats. The client of the caller shall, therefore, apply the following matching mechanism to determine whether an incoming request relates to the ongoing call:

1. If both the destination identity of the telephony call and the originator identity of the in-call incoming request are phone numbers in international format, the client of the caller shall compare all digits of the provided numbers to determine whether they match.
2. If any of the identities is not in international format, the client of the caller shall apply an enhanced matching mechanism between the destination identity from the telephony call and the originator identity of the incoming request, e.g. by comparing the 7 digits starting from the end of the number. It is left to the client implementation to apply an even more enhanced matching algorithm to decrease the probability of false matches.

The client shall consider the identities to be in international format if

- For a CS or multimedia telephony outgoing call, the digits dialled or taken from the address book start with a “+”.
- For an incoming request, the P-Asserted-Identity of the SIP request contains either:
  - a tel URI starting with a “+” without phone-context i.e. a global number, or
  - a SIP URI with user part starting with a “+”, a user=phone parameter and without a phone-context parameter in the user part.

Examples:

The destination identity of the outgoing telephony call: **+447123456789** (display string for an international format number).
The originator identity of the incoming request: **+447123456789**
➔ Matching result: Successful

When the applied enhanced matching algorithm is based on the 7 digits starting from the end of the number:
The destination identity of the outgoing telephony call: 0712**3456789** (non-international format).
The originator identity of the incoming request: +44712**3456789**
➔ Matching result: Successful

#### 2.5.2.2.2 Callee: relating the ongoing call with in-call incoming SIP requests

The originator identity of the telephony call and the originator identity of any in-call incoming request may be provided in various formats both in the home networks and when roaming. The client of the callee shall therefore apply the following matching mechanism to determine whether an incoming request relates to the ongoing call:

1. If both the originator identity of the telephony call and the originator identity of the in-call incoming request are phone numbers in international format, the client of the callee shall compare all digits of the provided numbers to determine whether they match.
2. If any of the originator identities is not in international format, the client of the callee shall apply an enhanced matching mechanism between the originator identity from the telephony call and the originator identity of the incoming request, e.g. by comparing the 7 digits starting from the end of the number. It is left to the client implementation to apply an even more enhanced matching algorithm to decrease the probability of false matches.

The client shall consider the identities to be in international format if

- For a CS incoming call, the Type Of Number (TON) of the Calling Party BCD Number is set to “international” as defined in [3GPP TS 24.008].
- For a multimedia telephony incoming call, the P-Asserted-Identity of the SIP INVITE request contains either:
  - tel URI starting with a “+” without phone-context i.e. a global number or
  - SIP URI with user part starting with a “+”, a user=phone parameter and without a phone-context parameter in the user part.
- For an incoming request, the P-Asserted-Identity of the SIP request contains either:
  - a tel URI starting with a “+” without phone-context i.e. a global number or
  - a SIP URI with user part starting with a “+”, a user=phone parameter and without a phone-context parameter in the user part.

Examples:

The originator identity of the incoming telephony call: **+447123456789** (display string for an international format number).

The originator identity of the incoming request: **+447123456789**
➔ Matching result: Successful

When the applied enhanced matching algorithm is based on the 7 digits starting from the end of the number:
The originator identity of the incoming telephony call: 00644712**3456789** (non-international format).
The originator identity of the incoming request: +44712**3456789**
➔ Matching result: Successful

### 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 and Refer-To header field included in outgoing SIP requests for Group Chat.

If no SIP or tel URI is present, the RCS client shall use the MSISDN extracted from SIP signalling that was cached as specified in section 2.5.2.1 or below. If that is not available, 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].

If the target number is a non-international format telephone number, the RCS client shall be able to send it as tel URI 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>*). In this case, if the response provides in the P-Asserted-Identity header field, the MSISDN of the contact in international format, the client shall cache for that contact the MSISDN extracted from the response for a duration equal to the value configured for the CAPABILITY INFO EXPIRY client configuration parameter defined in section A.1.9.

#### 2.5.3.2 In-call SIP requests

##### 2.5.3.2.1 Caller: addressing SIP requests towards the callee

The destination identity of the telephony call that the caller dials or gets from the address book may be in various formats. The client of the caller shall, therefore, apply the following principles for addressing the callee when triggers in-call SIP requests:

1. If the destination identity of the telephony call is in international format, the client of the caller shall use this information for addressing in-call SIP requests towards the callee.
2. If the destination identity of the telephony call is not in international format, the client of the caller shall use geo-local numbering of the destination identity of the telephony call for addressing in-call SIP requests towards the callee. If the request fails, the client of the caller shall attempt to correlate the destination identity of the telephony

call with his local identity records acquired from incoming SIP requests received in a window prior to the call and/or during the call using an enhanced matching mechanism between the destination identity from the telephony call and the incoming SIP requests, e.g. by comparing the 7 digits starting from the end of the number. It is left to the client implementation to set the time length of the window and apply an even more enhanced matching algorithm to decrease the probability of false matches.

a) If there is successful matching, the client of the caller shall use the “matched” destination identity from his local identity records for addressing in-call SIP requests towards the callee.
b) If there is no successful matching, the client of the caller shall use the destination identity from the telephony call that the caller dials or gets from his address book for addressing in-call SIP requests towards the callee. The client of the caller shall continue applying the enhanced matching mechanism for any in-call incoming SIP request until it matches the destination identity from the telephony call with the originator identity from an in-call incoming SIP request. Once there is a successful matching, it shall from then on use the “matched” originator identity from the SIP request for addressing any future in-call SIP requests towards the callee.

The client shall consider the identities to be in international format if

- for a CS or multimedia telephony outgoing call, the digits dialled or taken from the address book start with a “+”.

Examples:

The destination identity of the outgoing telephony call: +447123456789 (display string for an international format number).
The client of the caller uses the destination identity from the telephony call for addressing in-call SIP requests towards the callee.
The enhanced matching mechanism does not apply.

The destination identity of the outgoing telephony call: 0712**3456789** (non-international format).
The client of the caller shall use geo-local numbering of the destination identity of the telephony call for addressing in-call SIP requests towards the callee: tel:07123456789;phone-context=geolocal.<homedomain>, where <homedomain> needs to be replaced with the home network domain name as configured by the device (as per section 2.2.3 of [PRD-IR.92]).
If the in-call SIP request fails, the client shall apply the enhanced matching mechanism.
The originator identity of incoming SIP request: +447123456789.
➔ Matching result: Successful

#### 2.5.3.2.2 Callee: addressing SIP requests towards the caller

The originator identity of the telephony call may be provided in various formats both in the home networks and when roaming. The client of the callee shall therefore apply the following principles for addressing the caller when triggers in-call SIP requests:

1. If the originator identity of the telephony call is in international format, the client of the callee shall use this information for addressing in-call SIP requests towards the caller.
2. If the originator identity of the telephony call is not in international format, the client of the callee shall use geo-local numbering of the originator identity of the telephony call for addressing in-call SIP requests towards the caller. If the request fails, the client of the callee shall attempt to correlate the originator identity of the telephony call with his local identity records acquired from incoming SIP requests received in a window prior to the call and/or during the call using an enhanced matching mechanism between the originator identity from the telephony call and the incoming SIP requests, e.g. by comparing the 7 digits starting from the end of the number. It is left to the client implementation to set the time length of the window and apply an even more enhanced matching algorithm to decrease the probability of false matches.

    a) If there is successful matching, the client of the callee shall use the “matched” originator identity from his local identity records for addressing in-call SIP requests towards the caller.
    b) If there is no successful matching, the client of the callee shall use the originator identity from the telephony call for addressing in-call SIP requests towards the caller. The client of the callee shall continue applying the enhanced matching mechanism for any in-call incoming SIP request until it matches the originator identity from the telephony call with the originator identity from an in-call incoming SIP request. Once there is a successful matching, it shall from then on use the “matched” originator identity from the SIP request for addressing any future in-call SIP requests towards the caller.

The client shall consider the identities to be in international format if

- for a CS incoming call, the Type Of Number (TON) of the Calling Party BCD Number is set to “international” as defined in [3GPP TS 24.008].
- for a multimedia telephony incoming call, the P-Asserted-Identity of the SIP INVITE request contains either:

    - a tel URI starting with a “+” without phone-context i.e. a global number or
    - a SIP URI with user part starting with a “+”, a user=phone parameter and without a phone-context parameter in the user part.

Examples:

> The originator identity of the incoming telephony call: +447123456789 (display string for an international format number).
> The client of the callee uses the originator identity from the telephony call for addressing in-call SIP requests towards the caller.

The enhanced matching mechanism does not apply.

When the applied enhanced matching algorithm is based on the 7 digits starting from the end of the number:
The originator identity of the incoming telephony call: 00644712**3456789** (non-international format).
The client of the callee shall use geo-local numbering of the originator identity of the telephony call for addressing in-call SIP requests towards the caller: tel: 006447123456789;phone-context=geolocal.<homedomain>, where <homedomain> needs to be replaced with the home network domain name as configured by the device (as per section 2.2.3 of [PRD-IR.92]).
If the in-call SIP request fails, the client shall apply the enhanced matching mechanism.
The originator identity of incoming SIP request: +44712**3456789.**
➔ Matching result: Successful

#### 2.5.3.3 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.

#### 2.5.3.4 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 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.

### 2.5.4 Addressing related to Chatbots

#### 2.5.4.1 Chatbot Service Identifier

A Chatbot shall be addressed using a SIP URI and may be addressed using a tel URI. When the client is addressing the Chatbot the Chatbot SIP URI should be used if available instead of the tel URI.

To discover the Chatbot service identifier for a tel URI, either

- capability discovery is used or
- if capability discovery is not used, the procedures for the 1-to-1 Messaging technology selection as defined in section 3.2.1 are used.

If a tel URI is used

- for capability discovery via SIP OPTIONS request, both the tel URI and SIP URI are returned to the client in the P-Asserted-Identity header field of the SIP 200 OK response.
- for capability discovery via Presence, the SIP URI is returned in the presence service ID tuple
- for a non-Chatbot 1-to-1 Messaging communication request being referred to a Chatbot Messaging communication as per section 3.2.1.1, both the tel URI and SIP URI are returned to the client in the P-Asserted-Identity of the SIP 403 Forbidden response.

For the SIP URI,

- the FQDN used for bots hosted on a third-party provider of a Chatbot Platform shall include the “botplatform” subdomain as shown in Table 5;
- the FQDN used for bots hosted on a Chatbot Platform hosted by a Service Provider
  - should include the “botplatform” subdomain as shown in Table 5, and
  - may follow the format botplatform.mnc<MNC>.mcc<MCC>.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]), to enable the traceability of Chatbot transactions among operators;
- the username part, shown as <bot_service_id_userpart> in Table 5, shall be set to the unique identifier assigned by the Chatbot Platform to the Chatbot. It shall be formatted according to the following definition:

```
bot_service_id_userpart = bot_identifier ["." bot_publisher_id]
bot_identifier    = 1* (bot_id_allowed)
bot_publisher_id  = 1* (bot_id_allowed / ".")
bot_id_allowed    = ALPHA / DIGIT / bot_id_unreserved / escaped_char
bot_id_unreserved = "-" / "_" / "!" / "~" / "*" / "'" / "(" / ")" /
                    "&" / "=" / "+" / "$" / "," / ";" / "?" / "/"
escaped_char      = "%" HEXDIG
```

  where the different parts shall be used as follows:

  - The `bot_identifier` part shall be an identifier for the Chatbot
  - The `bot_publisher_id` part is, as indicated, optional. If provided, it shall be a unique identifier provided by the Chatbot Platform for the party (e.g. a business) providing the Chatbot. This will allow parties in the signalling path to determine that different Chatbots are provided by the same party.

| sip:<bot_service_id_userpart>@botplatform.<botplatformdomain> |
|---|

**Table 5: Chatbot service ID format**

#### 2.5.4.2 Anonymized Address for user of a Chatbot

An anonymized address used to hide the URI of a User shall follow the following syntax:

- sip:<token>@<routable hostname>;user=rcstk

NOTE: According to [RFC3261], if the URI contains a semicolon, the URI must be enclosed in angle brackets (< and >).

The ‘rcstk’ value in the ‘user’ URI parameter indicates that the user is using a token to hide their identity.

How the <token> is created is left to the Service Provider’s implementation. The token value shall be unique in the <routable hostname> realm.

#### 2.5.4.3 Indication that anonymization was used for the User's identity

In order for the client to know if a token is used by a Chatbot to communicate with the user, the following URI parameter is defined:

- tk
  - Values
    - off : Anonymization is not used for the conversation
    - on : Anonymization is used for the conversation

  The ‘*tk’* URI parameter shall be added by the Anonymization Function (AF) to the Chatbot’s SIP URI transported by the P-Asserted-Identity in SIP requests and responses going towards the client. The value of the ‘*tk’* SIP parameter indicates to the client whether a token is used between the AF and the Chatbot in the conversation. If the value is set to ‘off’, then no token is used in the conversation.

Usage example of a Chatbot SIP URI indicating that the Chatbot is using the token:

```
<sip:mybot@botplatform.foodomain;tk=on>
```

NOTE: According to [RFC3261], if the URI contains a semicolon, the URI must be enclosed in angle brackets (< and >).

#### 2.5.4.4 Anonymized and Non-Anonymized Conversations

When comparing P-Asserted-Identity SIP URI values in incoming SIP requests/responses, and in Message Store synchronizations to existing Chatbot contacts in the address book, any added URI parameters shall be ignored by the client. This allows anonymized and non-anonymized messages to be interleaved in the same Chatbot conversation history. The client shall still use the URI parameters as defined in section 2.5.4.3 to indicate to the user which messages were part of an anonymized conversation and which were part of a non-anonymized conversation.

## 2.6 Capability and new user discovery mechanisms

### 2.6.1 Capability discovery

The capability or service discovery mechanism is a process which enhances service usability by allowing a user to understand the subset of RCS services available to access and/or communicate with their contacts, at certain points in time.

When available, 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.
  - This method requires a specific application server (Options-AS) in the network to provide multidevice support and, potentially, include optimisations.
- 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 Open Mobile Alliance (OMA) SIP for Instant Messaging and Presence Leveraging Extensions (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 a Presence Server and optionally (see section 2.6.1.2.4) a XDM server in the network.

The discovery mechanism that is to be used by the device is by using the configuration parameter CAPABILITY DISCOVERY MECHANISM (see Annex A section A.1.9).

The interoperability between the mechanisms is provided based on network interworking (section 2.6.1.4.1). 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.

#### 2.6.1.1 Capability discovery process through SIP OPTIONS message

This mechanism for capability discovery is based on the exchange of a SIP OPTIONS request as defined in [RFC3261], a peer-to-peer message exchanged between clients.

This mechanism is based on the use of tags corresponding to the different RCS services that are defined in section 2.6.1.3 that are 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 request.
- 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.

In RCS, the SIP OPTIONS request shall NOT contain a Session Description Protocol (SDP) body.

Next to the relevant tags defined in section 2.6.1.3, a device should also add to the Contact header field the same feature tags used at SIP Registration (see section 2.4.4) if not already included in the SIP OPTIONS request/response for capability exchange and if they are part of the capabilities supported by the device at this time.

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 9)<br>Returned when User B is an RCS user and is currently registered | User B remains an RCS user<br>The capabilities returned in the 200 OK response (using tags as described in Table 9) are considered as the current communication options with User B | User B is marked as an RCS user<br>The capabilities returned in the 200 OK response (using tags as described in section Table 9) are considered as the current communication options with User B |
| 200 OK not including any of the tags used by RCS services (see Table 9)<br>Returned when User B is registered, but not with an RCS client | User B is not considered as an RCS user any longer<br>Only the non-RCS communication services (e.g. voice calls, SMS, MMS, etc.) are indicated as available1 | No change in User B's status<br>Only the non-RCS communication services (e.g. voice calls, SMS, MMS, etc.) are indicated as available |
| 480 TEMPORARY UNAVAILABLE or<br>408 REQUEST TIMEOUT<br>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<br>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 | User B is not considered as an RCS user any longer<br>Only the non-RCS communication services (e.g. voice calls, SMS, MMS, etc.) are indicated as available | No change in User B's status<br>Only the non-RCS communication services (e.g. voice calls, SMS, MMS, etc.) are indicated as available |

[1] Note that this means that an AS like the OPTIONS-AS described in section 2.6.1.1.1 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.

| Response | User B was a known RCS user before | User B was not a known RCS user before |
|---|---|---|
| Any other Final response returned by the network | User B remains an RCS user with unchanged capabilities.<br>NOTE: The client treats the final response as described in [3GPP TS 24.229]. | No change in User B's status |

**Table 6: Options response handling**

This is illustrated in Figure 2:

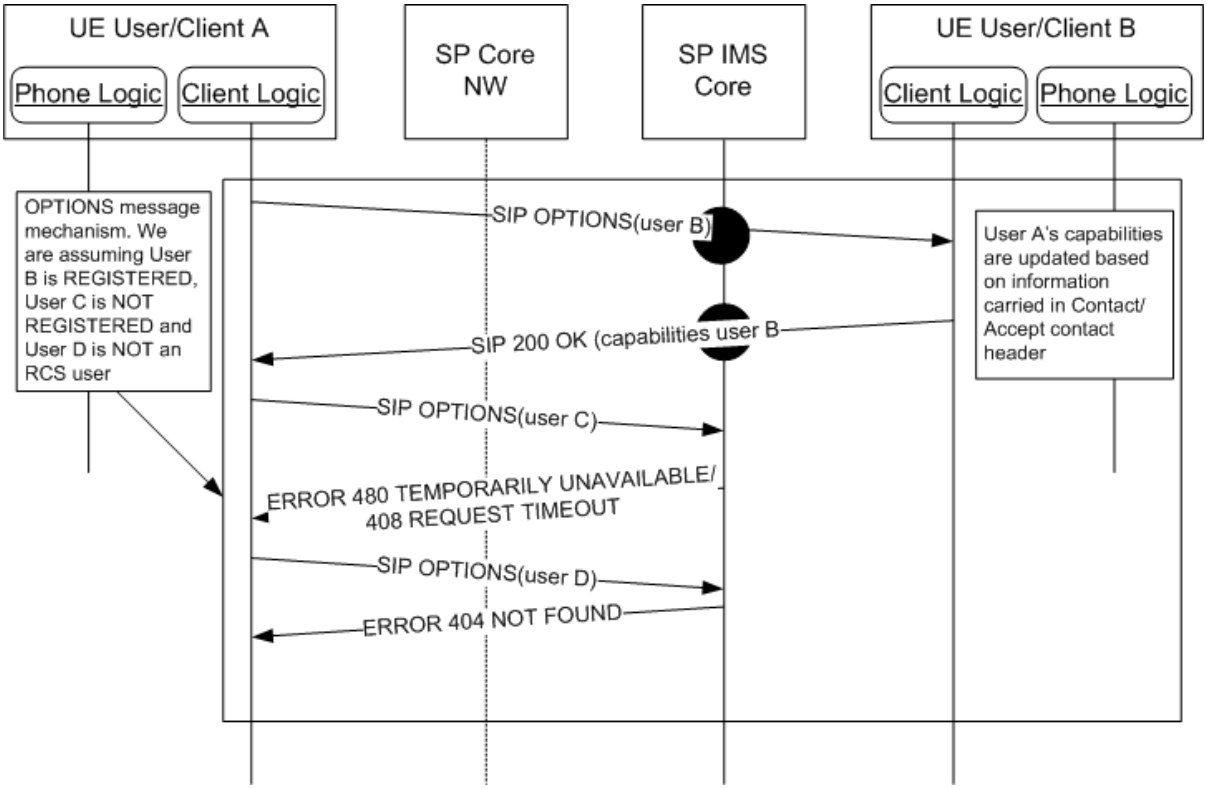



**Figure 2: Capabilities discovery via SIP OPTIONS message**

#### 2.6.1.1.1 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) and nothing specific is done, 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 as specified in section 2.10.2.
- Once the responses from the different devices are received, the AS will aggregate all the capabilities from the replies of the different clients and send them back to the requester.
- Even if not all of the replies have been received in less than a configurable amount of time, the AS will return the aggregated information received so far.

NOTE: the recommendation is to set the value to optimise the User Experience (UX) on the terminal.

- 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.

This is illustrated in Figure 3:

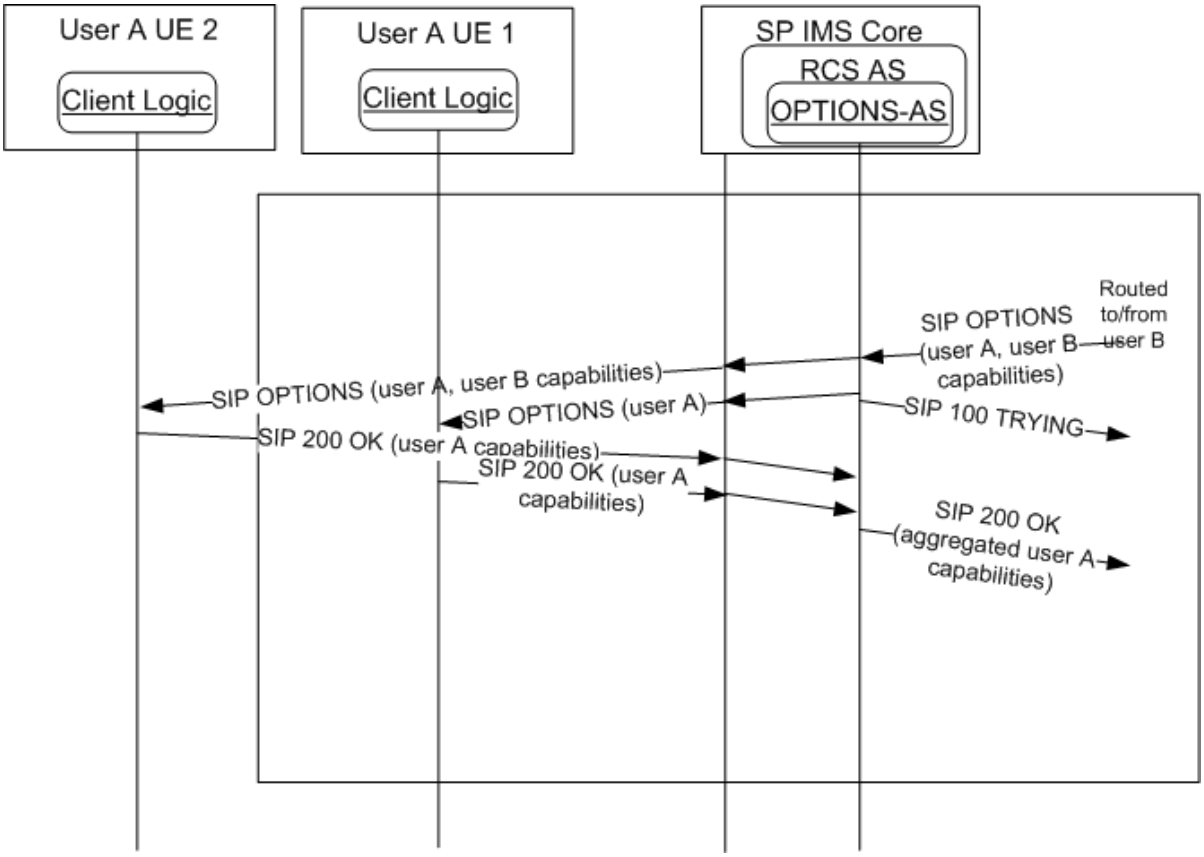



**Figure 3: Options application server: Capability aggregation on SIP OPTIONS request**

### 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 realised using the "*Service*" part of the Presence Data Model that 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). This SIP PUBLISH request shall not include an *Expires* header field. Service capabilities publication through OMA Presence Enabler [Presence2.0_TS] or [Presence] must follow [PDE_14] rules.

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

If changes are required in the published capabilities (for example, due to the behaviour specified in section 2.6.1.3), 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 Authorisation rules (see section 2.6.1.2.4) allow him to do so.

When using Presence as the enabler for Capability exchange, RCS clients shall retrieve the service capability information of contacts by means of Anonymous Fetch operations (as described in section 7.1 of [PRESENCE]). This will result in a single NOTIFY request indicating the service capabilities of that contact. The contact shall be considered as an RCS user only if the response includes one of the service-IDs described in Table 9. This information shall then be cached in the client as described in section 2.6.2.

If an RLS-URI (Resource List Server URI, see Annex A section A.1.2.1) 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 Authorisation for capabilities retrieval

To provide authorisation to retrieve the capabilities using an Anonymous Fetch request, an RCS Service Provider supporting the capability exchange using presence shall either provide a service provider policy on the presence server allowing anonymous subscriptions to retrieve the capabilities or set for every RCS subscriber a presence rules document in the presence XDMS providing such authorisation.

#### 2.6.1.2.5 Service part of the presence Data Model

A service capability is provided according to the model described in Table 7:

| 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:<br><presence> -> <tuple> | [RFC3863] and [Presence2.0_DDS] | According to the presence schema defined in the [Presence], services are presented with *tuple* elements. |
| Status<br><tuple> -> <status> -> <basic> -> Open | [RFC3863] and [Presence2.0_DDS] | Mandatory element in [RFC3863].<br>For every a tuple element that is published the value 'open' shall be used. It does not have any particular meaning in RCS context. |

| Attribute | Specification | Comment |
|---|---|---|
| Service-id<br><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. The Service-IDs that are used for the different services that are part of RCS are described in section 2.6.1.3. |
| Version<br><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). The Version that are used for the different services that are part of RCS are described in section 2.6.1.3. |
| Media<br><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 for which this is mentioned in section 2.6.1.3. |
| Contact<br><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.<br>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)<br>NOTE1: According to [RFC3863], "tuples that contain a <*basic*> element SHOULD contain a <*contact*> address".<br>Therefore, as a default- the <*contact*> element should be populated with a tel URI provided that:<br>The device has received a tel URI in P-Associated-URI header of 200 OK response to REGISTER request.<br>The service in question can utilise tel URIs. |
| Timestamp:<br><tuple> -> <timestamp> | [RFC3863] and [Presence2.0_DDS] | Timestamp when the presence information was published. |

**Table 7: Attributes of the Presence Service element**

### 2.6.1.3 Service/capability indicators

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 General Packet Radio Service (GPRS) is not possible with the used enablers.

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 8 assumes that all the user's devices have been configured with all the RCS services enabled and the network supports all the RCS services.

| Service | TERMINAL and STATUS REQUIREMENTS | Data Bearer | | | | | |
|---|---|---|---|---|---|---|---|
| | | 2G | EDGE | 3G | HSPA | LTE | Wi-Fi |
| Chat (1-to-1 or group) | None | Y | Y | Y | Y | Y | Y |
| File Transfer via HTTP | The relevant configuration parameters are correctly set | Y | Y | Y | Y | Y | Y |
| File Transfer via SMS | The relevant configuration parameters are correctly set | Y | Y | Y | Y | Y | Y |
| IP Voice Call [PRD-IR.92]/[PRD-IR.51] | N/A | N | N | N | N | Y (IR.92) | Y (IR.51) |
| IP Video Call [PRD-IR.94] | Support video profile (encoding /decoding). | N | N | N | N | Y (IR.94) | Y (IR.51/IR.94) |
| RCS IP Voice Call | N/A | N | N | Y[2] | Y[2] | Y[2] | Y[2] |
| RCS IP Video Call | Support video profile (encoding /decoding). | N | N | Y[2] | Y[2] | Y[2] | Y[2] |

[2] Only for devices not enabled for VoLTE or VoWiFi and 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<br>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 PUSH via SMS | The relevant configuration parameters are correctly set | Y | Y | Y | Y | Y | Y |
| Call Composer via Enriched Calling session | The relevant configuration parameters are correctly set | Y[3] | Y[3] | Y | Y | Y | Y |
| Call Composer via Multimedia Telephony session | The relevant configuration parameters are correctly set. | N | N | N | N | Y (IR.92) | Y (IR.51) |
| Post-Call | Support audio message profile (encoding /decoding). | Y | Y | Y | Y | Y | Y |
| Shared Map | The terminal should be on an active call[4] with the user the map is willing to be shared with. It is not available in multiparty calls. | Y[3] | Y[3] | Y | Y | Y | Y |
| Shared Sketch | The terminal should be on an active call[4] with the user the canvas is willing to be shared with. It is not available in multiparty calls. | Y[3] | Y[3] | Y | Y | Y | Y |
| Chatbot communication | The relevant configuration parameters are correctly set | Y | Y | Y | Y | Y | Y |

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

[4] In this context, the term active call is used to indicate that a voice call is taking place with the user the content 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.

| Service | TERMINAL and STATUS REQUIREMENTS | Data Bearer | | | | | |
|---|---|---|---|---|---|---|---|
| | | 2G | EDGE | 3G | HSPA | LTE | Wi-Fi |
| Uni-directional Plug-ins | The requirements of the underlying selected service apply | Y | Y | Y | Y | Y | Y |

**Table 8: RCS services: Terminal, status and data bearer requirements**

Table 9 lists the feature tags and Service IDs that are used for indicating that a specific RCS service is available:

| RCS service | | Tag |
|---|---|---|
| Chat | Tag | +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcse.im";+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.oma.cpm.session"<br>The RCS Client shall assume that Chat is available if either of the two tags mentioned is present. |
| | Service ID | Service-id: org.openmobilealliance:IM-session<br>Version: 1.0<br>Contact address type: tel / SIP URI<br>Or<br>Service-id: org.openmobilealliance:ChatSession<br>Version: 2.0<br>Contact address type: tel / SIP URI |
| File Transfer | Tag | +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.fthttp" |
| | Service ID | Service-id: org.openmobilealliance:File-Transfer-HTTP<br>Version: 1.0<br>Contact address type: tel / SIP URI |
| File Transfer via SMS | Tag | +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.ftsms" |
| | Service ID | Service-id: org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.ftsms<br>Version: 1.0<br>Contact address type: tel / SIP URI |
| IP video call | Tag | +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel";video |
| | Service ID | Service-id: org.3gpp.urn:urn-7:3gpp-service.ims.icsi.mmtel<br>Version: 1.0<br>Media capabilities: audio, video, duplex<br>Contact address type: tel/ SIP URI |
| Geolocation PUSH | Tag | +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.geopush" |
| | Service ID | Service-id: org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.geopush<br>Version: 1.0<br>Contact address type: tel/ SIP URI |

| RCS service | | Tag |
|---|---|---|
| Geolocation PUSH via SMS | Tag | +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.geosms" |
| | Service ID | Service-id: org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.geosms<br>Version: 1.0<br>Contact address type: tel / SIP URI |
| Call composer via Enriched Calling session | Tag | +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.gsma.callcomposer" |
| | Service ID | Service-id: org.3gpp.urn:urn-7:3gpp-service.ims.icsi.gsma.callcomposer<br>Version: 1.0<br>Contact address type: tel / SIP URI |
| Call composer via Multimedia Telephony session | Tag | +g.gsma.callcomposer |
| | Service ID | Service-id: org.3gpp.urn:urn-7:3gpp-service.ims.icsi.gsma.callcomposer<br>Version: 2.0<br>Contact address type: tel / SIP URI |
| Post-Call | Tag | +g.3gpp.icsi-ref=”urn%3Aurn-7%3A3gpp-service.ims.icsi.gsma.callunanswered” |
| | Service ID | Service-id: org.3gpp.urn:urn-7:3gpp-service.ims.icsi.gsma.callunanswered<br>Version: 1.0<br>Contact address type: tel / SIP URI |
| Shared Map | Tag | +g.3gpp.icsi-ref=”urn%3Aurn-7%3A3gpp-service.ims.icsi.gsma.sharedmap” |
| | Service ID | Service-id: org.3gpp.urn:urn-7:3gpp-service.ims.icsi.gsma.sharedmap<br>Version: 1.0<br>Contact address type: tel / SIP URI |
| Shared Sketch | Tag | +g.3gpp.icsi-ref=”urn%3Aurn-7%3A3gpp-service.ims.icsi.gsma.sharedsketch” |
| | Service ID | Service-id: org.3gpp.urn:urn-7:3gpp-service.ims.icsi.gsma.sharedsketch<br>Version: 1.0<br>Contact address type: tel / SIP URI |
| Chatbot Communication using sessions | Tag | +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.chatbot";+g.gsma.rcs.botversion="#=1" |
| | Service ID | Service-id: org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.chatbot<br>Version: 1.0<br>Contact address type: tel / SIP URI |
| Chatbot Communication using | Tag | +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.chatbot.sa";+g.gsma.rcs.botversion="#=1" |

| RCS service | | Tag |
|---|---|---|
| Standalone Messaging | Service ID | Service-id: org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.chatbot.sa<br>Version: 1.0<br>Contact address type: tel / SIP URI |
| Chatbot role | Tag | +g.gsma.rcs.isbot |
| | Service ID | Service-id: org.gsma.rcs.isbot<br>Version: 1.0<br>Contact address type: tel / SIP URI |
| Plug-ins | Tag | +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.plugin" |
| | Service ID | Service-id: org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.plugin<br>Version: 1.0<br>Contact address type: tel / SIP URI |

**Table 9: Complete SIP OPTIONS tag and Presence Service ID usage for RCS**

NOTE: 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. It should also be noted that in some cases, the tags employed in the SIP OPTIONS exchange match the standard tags.

When used in SIP OPTIONS exchanges these capabilities relating to In-Call Services (Shared Map, Shared Sketch) shall only be sent during an active call and shall be included only if the exchange takes place between the users in the active call.

Finally, when several IARI tag values or several ICSI tag values are included in a SIP OPTIONS request, consistently with [RFC3840], IARI tag values or ICSI tag values shall be concatenated using commas as shown 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 10: IARI tag concatenation format example**

#### 2.6.1.3.1 Future extensions to the mechanism

In addition to the aforementioned services 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 and Service ID format is defined.

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

For this purpose, following Capability Identifiers have been identified:

| RCS service | | Tag |
|---|---|---|
| Service Provider specific service | Tag | +g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.mnc<mnc>.mcc<mcc>.<service name>" |
| | Service-ID (based on IARI) | Service-Id: org.3gpp.urn:urn-7:3gpp-application.ims.iari.rcs.mnc<mnc>.mcc<mcc>.<service name><br>Version: Service Provider choice |
| | Service-ID (based on OMA scheme) | Service-Id: org.openmobilealliance:<RCS service name>.mnc<mnc>.mcc<mcc>.<service extension><br>Version: Service Provider choice |

**Table 11: Feature Tag and Presence service tuple proposal for future lines of work**

When using the OMA scheme for the Service ID, Service extension patterns including “mnc<mnc>.mcc<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).

For Service Provider Extensions, 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.

Examples of service extensions for Service Providers:

- OPTIONS tags:
  - +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"
- Service-id for extension(s)
  - to Chat with the OMA scheme:
    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.sfgroupchatMyFlavor

#### 2.6.1.3.2 Chatbot indicators

##### 2.6.1.3.2.1 Chatbot indicators for SIP OPTIONS

A Chatbot Platform supporting capability exchange via SIP OPTIONS:

- shall not include the Chat IARI value as defined in Table 9 in any SIP OPTIONS request or response that it generates to avoid interaction with other RCS services;
- shall include in any SIP OPTIONS request or response that it generates :

- the Chatbot role as defined in section 3.6.2.3;
- the Chatbot IARI value as defined in section 3.6.2.1;
- the Chatbot application version as defined in section 3.6.2.2.

A client that supports the Chatbot service shall include the Chatbot application related feature tags as defined in section 3.6.2.1 and section 3.6.2.2 during capability discovery via SIP OPTIONS.

Non-normative examples:

- Contact header of a SIP OPTIONS request or response sent from the Chatbot Platform:

```
Contact:<sip:foo.bar@botplatform.domain;+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.chatbot";+g.gsma.rcs.botversion="#=1";+g.gsma.rcs.isbot
```

- Contact header of a SIP OPTIONS request or response sent from the Chatbot Platform with support for both Chatbot Chat Session and Chatbot Standalone Messaging:

```
Contact:<sip:foo.bar@botplatform.domain;+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.chatbot,urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.chatbot.sa";+g.gsma.rcs.botversion="#=1";+g.gsma.rcs.isbot
```

- Contact header of a SIP OPTIONS request or response sent from the client:

```
Contact:<sip:foo.bar@domain>;+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.oma.cpm.session";+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.chatbot";+g.gsma.rcs.botversion="#=1"
```

- Contact header of a SIP OPTIONS request or response sent from a client that registered on a network that enabled only Chatbot Standalone Messaging:

```
Contact:<sip:foo.bar@domain>;+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.oma.cpm.msg";+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.chatbot.sa";+g.gsma.rcs.botversion="#=1"
```

#### 2.6.1.3.2.2 Chatbot indicators for Presence

In the NOTIFY request indicating the service capabilities of a Chatbot , a Chatbot Platform supporting capability exchange via presence:

- shall not include the service-id for Chat as defined in Table 9 to avoid interaction with other RCS services.
- shall include the service-id for Chatbot role as defined in Table 9.
- shall include the service-id for Chatbot Communication as defined in Table 9.

A client that supports the Chatbot service shall publish the Chatbot Communication Service as defined in Table 9.

#### 2.6.1.4 Interworking between the different mechanisms

##### 2.6.1.4.1 Coexistence between the discovery mechanisms via network interworking

When Service Providers use presence as the discovery mechanism, interoperability is achieved between such a Service Provider and those Service Providers who have selected SIP OPTIONS as the default discovery mechanism by bi-directional network based interworking.

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 4).
- Convert SIP ANONYMOUS SUBSCRIBE requests into SIP OPTIONS requests (Figure 5).

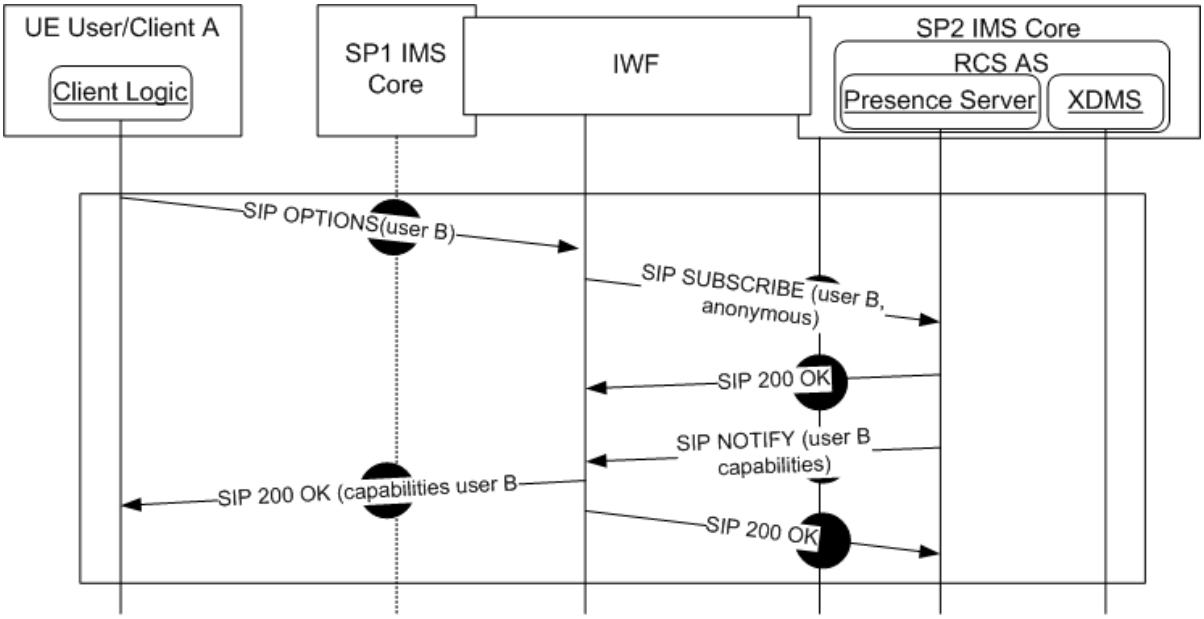



**Figure 4: Capability interworking via network: Options request**

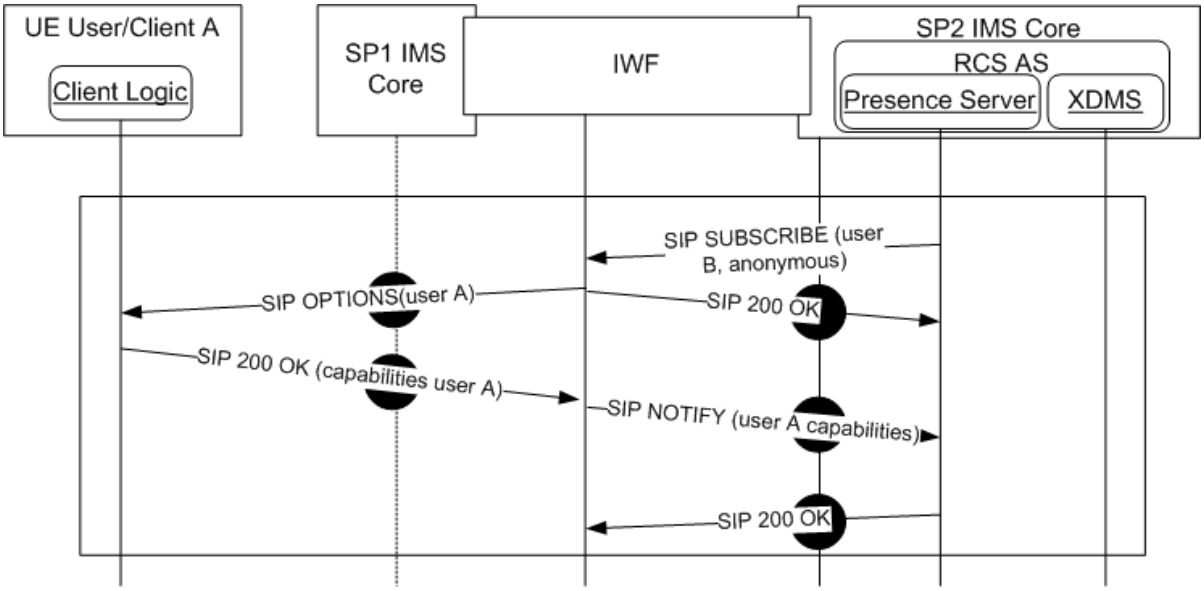



**Figure 5: Capability interworking via network: Presence request**

NOTE: Figure 4 and Figure 5 do not specify whether the IWF is deployed:

- In the Originating IMS network or
- In the Terminating IMS network or
- 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 specialised services that may be filtered from exposure to any external domains.
  - User based policy; including privacy or other subscriber level policies.

#### 2.6.1.4.2 Interworking with networks that disable Capability Discovery

When the Service Provider disables the capability discovery mechanism the network shall at the NNI still accept SIP OPTIONS requests for the capabilities of its users and return a SIP 200 OK including any of the agreed interworking service tags which are supported by User B as described in Table 9. If User B is currently not registered or is not a RCS user, the network shall respond in accordance with section 2.6.1.1 Capability discovery process through SIP OPTIONS message.

### 2.6.2 Handling of Capabilities

The enablers discussed in section 2.6.1 provide indications on the following

- capability information (i.e. whether a contact supports a service) and
- service availability information (i.e. whether a contact is currently likely in conditions that allow successful establishment of the service)

The Presence based mechanism provides this information using the presence service descriptions as described in section 2.6.1.2 which would indicate just service availability from which capability information can be derived (i.e. if a service is currently available for use with a contact, that contact is assumed to support the service). For SIP OPTIONS, a similar approach shall be followed, but there the use of the “automata” tag allows to indicate that a contact supports a service (i.e. capability information) without indicating service availability as described in section 2.6.1.1.

Capability information and service availability information obtained through the capability exchange enablers will be cached on the device. Within this cache, different expiry policies are applicable for capability information and service availability. The configuration parameters controlling this are described in section A.1.9.

When encountering an event related to a contact that does not correspond to the cached capability information of that contact, a client shall refresh those cached capabilities by initiating a Capability Discovery and Service Availability request. This shall be done when:

- A 1-to-1 SIP request to an RCS enabled contact results in a SIP 404 response or
- A Geolocation Push or File Transfer via HTTP request is received from a contact for which the corresponding capability wasn’t part of the cache or
- A SIP OPTIONS request carrying at least one RCS feature tag is received from a non-RCS contact.
- A contact initiates an enriched calling pre-call, post-call or in-call service for which the corresponding capability wasn’t part of the cache

NOTE: Reception of a Chat or Standalone Message from a non-RCS contact is not included because that could be the result of interworking. An IP Video Call from such a contact is excluded because that may come from a user that only supports IP Video Call.

#### 2.6.2.1 Service Provider Controlled Service Capabilities Handling

The following items can be configured subject to the Service Provider’s policies (see section A.1.9):

1. An expiry of the capabilities for a specific RCS or non-RCS contact through respectively the CAPABILITY INFO EXPIRY and the NON RCS CAPABILITY INFO EXPIRY client configuration parameters.

   NOTE: Chatbots are considered to be RCS contacts.

2. An expiry of the service availability information for a specific RCS contact through the SERVICE AVAILABILITY INFO EXPIRY client configuration parameter.
3. The Contacts with a telephone number-based address considered for the capability discovery depending on their prefix through the CAPABILITY DISCOVERY ALLOWED PREFIXES client configuration parameter.

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

## 2.7 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 affect 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]) |
| Hyper Text Transfer Protocol (HTTP) | XML configuration access protocol (XCAP) transactions<br>HTTP configuration mechanism<br>File Transfer<br>Access to Message Store Server<br>Chatbot information query<br>Service Provider Chatbot directory query<br>Retrieval of the lists of Chatbots requiring specific management<br>Plug-ins initial Catalog retrieval or refresh | TCP/IP | Hyper Text Transfer Protocol Secure (HTTPS) |

**Table 12: 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*", "*psSignallingRoaming*" and "*wifiSignalling*" in [PRD-RCC.15] section 2.2.2.2.

NOTE: The "psSignallingRoaming" parameter is defined as a temporary workaround to address PS roaming related issues identified in live deployments.

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 2.7.2 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 2.7.1.
- For HTTP, 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 protocol 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 2.12.

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 [PRD-RCC.15], section 2.2.2.2. Consequently, a Service Provider provides this information as part of the first-time or re-configuration scenarios described in section 2.3.

### 2.7.1 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 that 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 Real-Time Transport Control Protocol (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 RCS IP Video Calling (see section 3.5) 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.

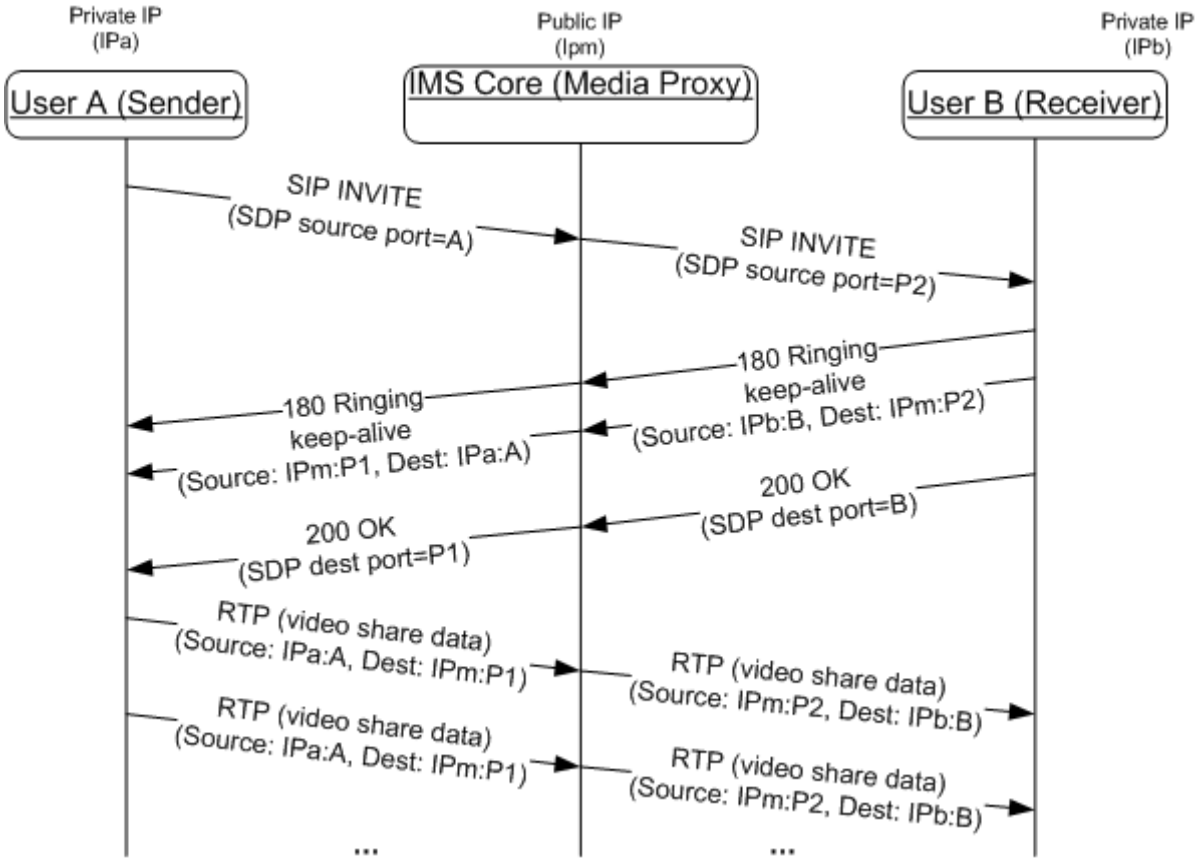



**Figure 6: RTP symmetric media path establishment**

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

### 2.7.2 MSRP session matching

For all services using MSRP, an RCS client shall set up MSRP sessions as per [RCS-CPM-CONVFUNC-ENDORS].

### 2.7.3 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.

## 2.8 RCS and Access Technologies

### 2.8.1 RCS and Cellular/EPC-integrated Wi-Fi Access

#### 2.8.1.1 Access used by RCS in relation to VoLTE/VoWiFi

A device providing all RCS services (i.e. including VoLTE/ViLTE or VoWiFi) shall support the procedures described in [PRD-NG.102]. These procedures are dependent on the RCS VOLTE SINGLE REGISTRATION configuration parameter defined in section A.1.6.

#### 2.8.1.2 LTE 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 (NGBR) services and GBR Monitoring Function are all as specified in [PRD-IR.92] for devices enabled for VoLTE. None of this is applicable to other devices and access networks other than Long Term Evolution (LTE).

#### 2.8.1.3 Bearer aspects

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

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

For all RCS IMS traffic, the following applies:

- For a device enabled for VoLTE: LTE QCI (QoS class identifier) 8 and 9 shall be supported so that either may be used for MSRP traffic.
- For a device using the IMS APN for RCS (see section 2.8.1.4): LTE QCI (QoS class identifier) 8 and 9 shall be supported so that either may be used for MSRP traffic.
- For other devices: no requirements.

#### 2.8.1.4 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 section 2.8 are applicable also in the roaming scenarios. Specific roaming considerations for the different RCS device types (as specified in section 2.2):

- All services on a primary device enabled for VoLTE, shall follow the general rules as per [PRD-IR.88], APN usage as per [PRD-NG.102].
- All services on a primary device enabled for VoWiFi shall follow the general rules as per [PRD-IR.61], APN usage as per [PRD-NG.102].
- Other devices: no specific requirements.

The APN to be used to access the RCS services[5] depends on the capacity of the device and the network to support an IMS APN as per [PRD-IR.88], on the device configuration and on the client type (see section 2.2):

- When the device is configured for VoLTE or VoWiFi, an embedded client shall use the APN indicated in [PRD-NG.102].
- For an embedded client on other devices, the behaviour shall depend on the setting of the RCS VOLTE SINGLE REGISTRATION configuration parameter (see section A.1.5):

NOTE: The RCS VOLTE SINGLE REGISTRATION configuration parameter is used in this case even if there is no VoLTE registration from the device because the required behaviour is similar.

  - The IMS APN shall be used to access the RCS services when the device is configured through the RCS VOLTE SINGLE REGISTRATION configuration parameter defined in section A.1.5 to use a single registration approach.
  - The Home Operator Services (HOS) APN shall be used to access the RCS services when the device is configured through the RCS VOLTE SINGLE

[5] 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.

REGISTRATION configuration parameter defined in section A.1.5 to use a dual registration approach.

NOTE: 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. If no IMS roaming agreement is in place, the visited network would degrade any requested QoS based on local configuration, as per normal procedures in the MME for any APN as specified in [PRD-IR.88]. The operator could optionally have a QoS data roaming agreement to ensure that QCI=5 for the IMS APN is allowed in the visited network and that either QCI=8 or QCI=9 for MSRP will be allowed on the IMS APN. If the appropriate QCI bearer cannot be set up for MSRP when on LTE access, MSRP traffic will be on the default QCI=5 bearer unless prevented by the terminal or 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.

For an embedded client, the APN to use for HTTP and XCAP shall be the HOS APN as defined in [PRD-IR.88]. The HOS APN can be configured to be, or by default be, the device's generic data access APN (i.e. the internet APN[6]). The network should use the HOS APN when providing the RCS client with its configuration, in order to prevent unwanted data charging for client provisioning traffic.

It is out of scope of this specification how the HOS APN is configured on the device.

- Downloadable clients shall use the internet APN.

#### 2.8.1.5 Data Off

Users can switch cellular data usage off locally on their device. To allow the operator to offer IR 92 / IR 94 and RCS services to their customers even in these use cases, the data off switch shall have an operator configurable impact on the device connectivity. The service provider should ensure a good service experience if IP service usage is allowed although the data switch was set to off by the end user. The procedures for cellular data off are applicable for primary devices when RCS is not using the internet APN (see section 2.8.1.4).

For services using IMS protocols, the implementation of cellular data off shall be based on 3GPP PS Data Off defined in [3GPP TS 24.229-rel15] with the following additional clarifications and requirements.

The cellular data off exempt is configured for RCS services via the following configuration parameters:

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

- RCS MESSAGING DATA OFF, FILE TRANSFER DATA OFF, CONTENT SHARE DATA OFF, PRE AND POST CALL DATA OFF as defined in section A.1.14.1,
- MMTEL_voice_exempt, MMTEL_voice_roaming_exempt as defined in [PRD-IR.92] and [3GPP TS 24.275],
- SMSoIP_exempt, SMSoIP_roaming_exempt as defined [PRD-IR.92] and [3GPP TS 24.167],
- MMTEL_video_exempt, MMTEL_video_roaming_exempt as defined in [PRD-IR.94] and [3GPP TS 24.275].

When registering in the network, the client shall include in the contact header of any SIP REGISTER request the "data off" feature tag defined in Table 3 indicating the data-off status in accordance with section B.3.1.0 of [3GPP TS 24.229-rel15].

If cellular data off is active and if for a given SIP registration (i.e. a single registration for all RCS service including VoLTE, ViLTE, SMS over IP and RCS services or a registration for RCS services only) no service is configured as a cellular data off exempt service, then the client shall not register in IMS.

If cellular data off is active and there are one or more services configured as cellular data off exempt services for a given SIP registration, then the client shall include in the Contact header of the SIP REGISTER only the service feature tags defined in Table 3 of the services being configured as cellular data off exempt services.

For a description of data off and service availability handling for VoLTE, ViLTE and SMS over IP refer to [PRD-IR.92] and [PRD-IR.94].

If cellular data off is active and there are one or more services configured as cellular data off exempt services for which capability discovery is required, then the client

- shall advertise capability information for services, including the ones not being cellular data off exempt services, and
- shall not advertise the service availability of services being not cellular data off exempt services.

For an overview of RCS services with capability information and service availability refer to section 3.3.1 of [PRD-RCC.71].

If cellular data off is active and there is no service configured as cellular data off exempt service for which capability discovery is required, then the client shall disable capability discovery.

For services using non-IMS protocols having no data connection at the time of disabling, no additional actions are required by the client. If disabled by Data off configuration the service will not be available at the time of invocation.

For services using non-IMS protocols and having a data connection active at the time of disabling, the ongoing session or transaction shall be terminated.

The procedures for the handling of cellular data off for the configuration procedures defined in section 2.3 are defined in [PRD-RCC.14].

The procedures for the handling of cellular data off for supplementary service management are defined in [PRD-IR.92].

#### 2.8.1.6 Summary of conditions and parameters that control the access network used

The combination of the switches, configuration parameters and coverage conditions introduced to control the connection through which the service is delivered leads to the behaviour described in Table 13:

| # | Telephony coverage | Wi-Fi Coverage | Registration approach following from RCS VOLTE SINGLE REGISTRATION[7][8] | RCS Service in Data Off[9] | Cellular Data Switch | Used network for RCS[10] |
|---|---|---|---|---|---|---|
| 1 | VoLTE | N/A | Single Registration | N/A | On | Cellular (IMS APN) |
| 2 | VoLTE | N/A | Single Registration | On | Off | Cellular[11](IMS APN) |
| 3 | VoLTE | N/A | Single Registration | Off | Off | None (RCS unavailable)[11] |
| 4 | CS | No | Single Registration | N/A | On | Cellular (IMS APN) |
| 5 | CS | No | Single Registration | On | Off | Cellular (IMS APN) |
| 6 | CS | No | Single Registration | Off | Off | None (RCS unavailable) |
| 7 | CS | Yes | Single Registration | N/A | N/A | Non EPC-integrated Wi-Fi |
| 8 | VoWiFi | Yes | Single Registration | N/A | N/A | EPC-integrated Wi-Fi |
| 9 | VoLTE or CS | No | Dual Registration | N/A | On | Cellular (HOS APN) |
| 10 | VoLTE or CS | No | Dual Registration | On | Off | Cellular (HOS APN) |

[7] See A.1.6.2

[8] Dual registration is used when the RCS VoLTE SINGLE REGISTRATION configuration parameter is configured to 0, when RCS VoLTE SINGLE REGISTRATION configuration parameter is configured to 2 and the device is roaming. Otherwise single registration is used.

[9] i.e. at least one of RCS MESSAGING DATA OFF, FILE TRANSFER DATA OFF, CONTENT SHARE DATA OFF and PRE AND POST CALL DATA OFF as defined in A.1.14 is set to 1 or it is set to 2 and the device is attached to the HPLMN or IP VIDEO CALL DATA OFF defined in A.1.14 is set to 1 or it is set to 2 and the device is attached to the HPLMN on a device that is not enabled for VoLTE.

[10] i.e. for traffic related to Standalone Messaging, 1-to-1 Chat, Group Chat, File Transfer, RCS IP Voice Call, RCS IP Video Call and Extension to Extension traffic as defined in section 3.2.2, 3.2.3, 3.2.4, 3.2.5, 3.4, 3.5 and **Error! Reference source not found.** respectively. VoLTE, ViLTE, SMSover IP and MMS always use the cellular network.

[11] Case assuming VoLTE remains switched on when data is off. If not, available cellular coverage is assumed to be CS.

| # | Telephony coverage | Wi-Fi Coverage | Registration approach following from RCS VOLTE SINGLE REGISTRATION[78] | RCS Service in Data Off[9] | Cellular Data Switch | Used network for RCS[10] |
|---|---|---|---|---|---|---|
| 11 | VoLTE or CS | No | Dual Registration | Off | Off | None (RCS unavailable) |
| 12 | VoLTE, VoWiFi or CS | Yes | Dual Registration | N/A | N/A | Non EPC-integrated Wi-Fi |
| 13 | None | No | N/A | N/A | N/A | None (RCS unavailable) |
| 14 | None | Yes | N/A | N/A | N/A | Non EPC- integrated Wi-Fi |

**Table 13: APN configuration proposal for data traffic and roaming**

### 2.8.2 Other access networks

In addition to the cellular PS access networks described in section 2.8.1, 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 2.4 details when such networks can be used and how to use RCS through them. Section 2.6.1.3 provides a guideline for which services can be used when connected through different types of access networks including broadband access.

## 2.9 End User Confirmation Requests

RCS clients shall support the End User Confirmation Request enabler as defined in [PRD-RCC.15] section 3.1.

## 2.10 Multidevice support

### 2.10.1 Overview

As shown in section 2.8.2, 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. This multidevice environment allows a user to answer a call or respond to a message from a device/client that suits their purpose.

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. 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 Serving Call Session Control Function (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 Decline then the S-CSCF should set the *cause=603* along with an optional protocol-text (e.g. *SIP;cause=603;text="Decline"*), 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.

### 2.10.2 Addressing of individual clients

Any Application Server (e.g. a Messaging Server or OPTIONS AS) can address an individual RCS client using information received with the third party registration (using the sip.instance feature tag).

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 where including a Contact header is possible, to allow identification by an application server.

## 2.11 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].

## 2.12 Access Security

### 2.12.1 IMS Security

#### 2.12.1.1 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.

#### 2.12.1.1.1 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]).

#### 2.12.1.1.2 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).

In RCS, a client is enabled for SIP Digest authentication by the Service Provider via the client configuration parameter IMS Mode Authentication Type defined in section 2.2.1.2 of [PRD-RCC.15].

When an RCS client is enabled for SIP Digest authentication, then the client shall use the configuration parameters defined in section 2.1.2.2 of [PRD-RCC.15] and section A.1.6.1 of the current document to populate the Authorization header field parameters for authentication in accordance the rules and procedures of section 5.1.1 of [3GPP TS 24.229] for IMS registrations using SIP Digest as follows:

- The private user identity for the "username" Authorization header field parameter shall be taken from the value of the configuration parameter Realm User Name defined in section 2.2.1.2 of [PRD-RCC.15].
- The domain name of the home network for the "realm" Authorization header field parameter shall be taken from the value of the configuration parameter Realm defined in section 2.2.1.2 of [PRD-RCC.15].
- The SIP URI of the domain name of the home network for the "uri" Authorization header field directive shall be taken from the value of the configuration parameter Home_network_domain_name defined in section A.1.6.1.
- For the initial registration using SIP Digest the client shall populate the value of the Authorization header field "response" as defined in section 5.1.1.2.3 of [3GPP TS 24.229] using

    - the username taken from the value of the configuration parameter Realm User Name defined in section 2.2.1.2 of [PRD-RCC.15],
    - the password taken from the value of the configuration parameter Realm User Password defined in section 2.2.1.2 of [PRD-RCC.15],
    - the realm taken from the value of the configuration parameter Realm defined in section 2.2.1.2 of [PRD-RCC.15],

- the URI taken from the value of the configuration parameter Home_network_domain_name defined in section A.1.6.1.

The IMS registration flow for SIP digest authentication is shown in Figure 7. In this example flow, the RCS client is connected to the IMS core over a Wi-Fi internet broadband connection.

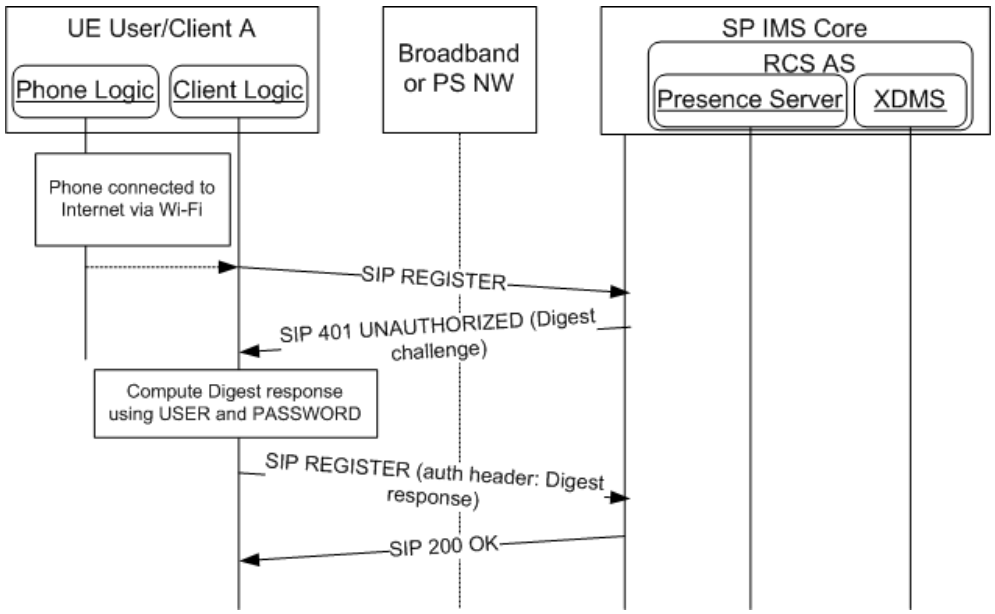



**Figure 7: Registration with SIP Digest Authentication**

If digest authentication fails two times with a SIP 401 UNAUTHORISED 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.

If the client is enabled for the SIP Digest authentication type, then the use of SIP/TLS is configured by the Service Provider via the access network specific transport signalling configuration parameters defined in section 2.2.1.2 of [PRD-RCC.15].

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: 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 [RFC2716] 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.

#### 2.12.1.2 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/VoWiFi | Cellular PS Access | SIP Digest (with or without TLS) or IMS AKA with IPsec | IMS AKA with IPSec may be used when supported by both device and the network.<br>SIP Digest with or without TLS is used depending on (pre-)configuration |
| | access over EPC-integrated Wi-Fi | SIP Digest (with or without TLS) or IMS AKA with IPsec | IMS AKA with IPSec may be used when supported by both device and the network.<br>SIP Digest with or without TLS is used in cases when pre-configured |

| Device | Access | Applicable Security Methods | Applicability and Suitability |
|---|---|---|---|
| | 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<br>SIP Digest with or without TLS is used in cases when pre-configured or when the mobile device does not support IMS AKA for WLAN access |
| VoLTE/VoWiFi configured mobile client | Cellular PS Access | SIP Digest, SIP Digest with TLS or IMS AKA with IPsec[12] as specified in [PRD-NG.102]<br><br>NOTE: The configuration to any other method is not possible. | AKA credentials stored securely in a UICC such as an xSIM. The method used depends on the relation to the registration for VoLTE/VoWiFi (see [PRD-NG.102]) |
| | EPC-integrated Wi-Fi | SIP Digest, SIP Digest with TLS or IMS AKA with IPsec[12] as specified in [PRD-NG.102]<br><br>NOTE: The configuration to any other method is not possible. | AKA credentials stored securely in a UICC such as an xSIM. The method used depends on the relation to the registration for VoLTE/VoWiFi (see [PRD-NG.102]) |
| | Non-cellular broadband (Wi-Fi) access | SIP Digest, SIP Digest with TLS or IMS AKA with IPsec[12]. | SIP Digest with TLS is recommended over SIP Digest without TLS<br><br>SIP Digest with or without TLS is used in cases when pre-configured or when the mobile device does not support IMS AKA for WLAN access. |
| Broadband Access Enabled | | SIP Digest or SIP Digest with TLS | SIP Digest with TLS is recommended over SIP Digest without TLS<br><br>SIP Digest is used for mobile devices which do not support IMS AKA for WLAN access. |

**Table 14: Access Signalling Security Profiles for RCS**

[12] 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 2.2.1.1.2 and 2.2.2.1.3 of [PRD-RCC.15], 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).

#### 2.12.1.3 Access Media Security

##### 2.12.1.3.1 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: [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 [PRD-RCC.15]), 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 be 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: 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: 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.

#### 2.12.1.3.2 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 [PRD-RCC.15], 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 2.7) 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.

### 2.12.2 OpenID Connect

#### 2.12.2.1 Overview

OpenID Connect provides a generic solution for user authentication independent from the type of device and the access network. It encompasses authentication, authorization and consent management for primary and secondary devices.

The procedures of OpenID Connect shall be supported by the client for all RCS services and cross service functionality using HTTP as access technology, i.e. for

- the client configuration based on the procedures defined in [PRD-RCC.14]
- File Transfer as defined in section 3.2.5
- the access to the Common Message Store as defined in section 2.12.3

#### 2.12.2.2 OpenID Connect Authentication Flow

The RCS client shall support the procedures of the OpenID Connect Authentication Flow using the Authorization Code Flow as defined in [OpenID Connect] as follows.

If the client has sent a HTTP request to an endpoint in the network providing one of the RCS services or cross service functionality above and the network endpoint returns a HTTP 302 Found response, then the client shall use the value of the HTTP Location header to connect to the Open ID Connect authorization endpoint. This request forms the OpenID Connect authentication request.

On reception of the OpenID Connect authentication request the OpenID Connect authorization endpoint performs the procedures for user authentication, authorization and consent.

If the procedures for authentication, authorization and consent succeed or fail, then the OpenID Connect authorization endpoint returns a HTTP 302 Found response. The client shall use the value of the HTTP Location header to connect to the network endpoint. This request forms the OpenID Connect authentication response.

On reception of the OpenID Connect authentication response the network endpoint performs the procedures as defined for the RCS service or cross service functionality. In

result of the processing, the network endpoint returns a success or failure response to the client as defined for the respective RCS service or cross service functionality.

The network endpoint and the OpenID Connect authorization endpoint shall preserve the values of the client's initial HTTP request during the OpenID Connect authentication request. The path and query components of the URIs used for the OpenID Connect authentication request procedure are opaque for the client.

The RCS client shall support for the OpenID Connect authentication flow the procedures for HTTP state management defined in [RFC6265]. This allows an OpenID Connect authentication endpoint to return in HTTP responses a Set-Cookie header. The client shall apply the parsing and storage procedures of the Set-Cookie header as defined [RFC6265]. It shall send the cookie header in HTTP requests to the OpenID Connect authentication endpoint respecting the cookie attributes in the Set-Cookie header in accordance with [RFC6265].

#### 2.12.2.3 Authentication, authorisation and user consent methods for OpenID Connect

The RCS client shall support the methods for user authentication, authorisation and user consent for OpenID Connect as defined for the Service Provider Device Configuration defined in [PRD-RCC.14].

In addition, if the client supports the procedures for authentication using a bootstrapped security association as defined in [3GPP TS 24.109], then the OpenID Authentication endpoint shall be able to invoke the authentication procedure accordingly.

### 2.12.3 Common Message Store Authentication and Security

The RCS client shall support the authentication and security mechanisms described in [CPM-MSGSTOR-REST] for access to the Common Message Store.

Authentication to the Message Store using RESTful API calls shall use one of the following methods:

1. Authentication using a bootstrapped security association as defined in section 4.1.2
2. HTTP Digest authentication as defined in section 4.1.4
3. HTTP Basic authentication as defined in section 4.1.5
4. OpenID Connect based authentication as defined in section 2.12.2 and 4.1.3

## 2.13 Emergency Services

### 2.13.1 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] in section 5.2 and [PRD-IR.51] in section 5.3 specify Emergency Services support.

### 2.13.2 RCS Service Feature List

Emergency Services support is provided in the following RCS Service Feature:

- 1-to-1 Chat

## 2.14 CPIM header extension support

RCS service definitions make use of the CPIM header extensibility mechanism defined in section 3.4 of [RFC3862] beyond the header namespace defined in section 6.1 of [RFC5438]. The CPIM header extension support mechanism ensures interoperability with messaging clients not supporting these extensions.

### 2.14.1 CPIM header extension support feature tag

The CPIM header extension support feature tag is the indication of an entity to support the procedures of CPIM header extensibility defined in this section.

The indication of the CPIM header extension support is provided by a feature parameter as defined in section 9 of [RFC3840]. The feature parameter is encoded as an "other-tags" feature tag in accordance with the definitions of section 9 of [RFC3840].

The feature tag name shall be set to "g.gsma.rcs.cpimext". The media feature tag shall have no value.

Security considerations for this media feature tag are discussed in section 11.1 of [RFC3840].

The indication of the CPIM header extension support is represented as follows:

+g.gsma.rcs.cpimext

### 2.14.2 Procedures in the client

A client supporting CPIM header extensions shall advertise its support by means of the feature tag defined in section 2.14.1 in the Contact header of

- SIP REGISTER requests and
- SIP INVITE requests and 200 OK responses for
    - Chat,
    - Chatbot sessions,
    - Group Chat,
    - Large Message Mode sessions and
    - Call Composer sessions as defined in [PRD-RCC.20].

A client supporting CPIM header extensions shall, on reception of a CPIM message, check the value of the CPIM "NS" headers contained in the message.

If

- the URI value and the associated namespace of a given CPIM "NS" header is known to the client and
- a header name containing the associated name prefix is known to the client,

then the client shall process the CPIM header in accordance with the definitions of the extension header.

If

- the URI value and the associated namespace of a given CPIM "NS" header is known to the client and
- a header name containing the associated name prefix is not known to the client,

then the client shall ignore the CPIM header.

If the URI value and the associated namespace of a given CPIM "NS" header is unknown to the client, then the client shall ignore all CPIM headers containing the associated name prefix.

### 2.14.3 Procedures in the Messaging Server

#### 2.14.3.1 Session Initiation

An originating Messaging Server supporting the media plane handling of CPIM header extensions defined in section 2.14.3.2 shall advertise this by including the feature tag defined in section 2.14.1 in the Contact header field of all SIP INVITE requests for a Chat, Group Chat or Standalone Message session.

A terminating Messaging Server supporting the media plane handling of CPIM header extensions defined in section 2.14.3.2 shall advertise this by including the feature tag defined in section 2.14.1 in the Contact header field of the 200 OK response to a SIP INVITE for a Chat, Group Chat or Standalone Message session.

A Controlling Function supporting the media plane handling of CPIM header extensions defined in section 2.14.3.2 shall advertise this by including the feature tag defined in section 2.14.1 in the Contact header of the SIP INVITE for Group Chat or Standalone Message session and in the 200 OK response to a SIP INVITE for a Group Chat or Standalone Message session.

#### 2.14.3.2 Media Plane Handling

The Messaging Server sending a CPIM message shall detect whether CPIM header extensions are supported for the Chat, Group Chat or Standalone Messaging session. CPIM header extensions are supported for a session, if the Messaging Server received the CPIM header extension support feature tag defined in section 2.14.1 in the Contact header of the SIP INVITE request or 200 OK response to establish the Chat, Group Chat or Standalone Messaging session.

If for the session the CPIM header extension is not supported and if the Messaging Server needs to send a CPIM message in such a session, then the Messaging Server shall parse the CPIM "NS" headers contained in the message.

If a given CPIM "NS" header contains a URI value different from the URI value defined in section 6.1 of [RFC5438] and there is no other compatibility rule defined for the namespace or extension header in the corresponding service definitions, then when forwarding the CPIM Message the Messaging Server

- shall remove the "NS" header, and
- shall remove all CPIM headers containing the associated name prefix.

NOTE: an example of such other compatibility rules is provided in section 3.6.8.7.

If all CPIM "NS" headers have been processed, then the Messaging Server shall continue processing with the resulting CPIM message as defined for the corresponding RCS service.

# 3 RCS Services

## 3.1 General Service Overview

RCS provides several services that fit into the framework defined in section 2.

Section 3.2 describes messaging services and introduces services that enhance the user's messaging experience. Section 3.2.1 describes 1-to-1 Messaging and the selection of different technologies to provide this service. Section 3.2.2 describes the standalone messaging service based on OMA CPM that is considered as an evolution of the SMS/MMS messaging services providing fewer restrictions, allowing 1-to-Many Messaging and providing the interworking capability with those services. Section 3.2.3 introduces the 1-to-1 chat service that provides a more real-time experience through "isComposing" indications in addition to the store and forward functionality, including delivery and display notifications, that allows reaching users while they are offline. Section 3.2.4 describes the Group Chat service which provides multiparty scenarios. Section 3.2.5 describes the File Transfer service allowing a user to exchange any type of file with another user. A Geolocation Push service is introduced in section 3.2.6 which allows a user to share their location (or any other desired location) with another user. Section 3.2.7 describes the Audio Messaging service allowing a user to share an audio file with another user. Section 3.2.8 describes the procedures on sharing Uni-directional Plug-in content with clients that may have or may not have the Plug-in installed.

Section 3.3 introduces the Content Sharing services allowing the user to share multi-media content ahead of the call to provide context to the called party when the call is set up. The content sharing services also includes the capability to exchange a map or a drawing canvas in real-time with another user during a voice call. In other circumstances, the File Transfer service or the messaging service could be used. It also allows sharing a note (reason) or a voice message after an unanswered call.

Section 3.4 and 3.5 describe respectively an IP based voice and video call functionality for broadband access clients and mobile devices. These services include support for a set of supplementary services and ensure the quality of service delivery when used on EPC-integrated Wi-Fi and LTE access. For the voice call, a mobile device on EPC-integrated Wi-Fi and LTE provides continuity to a CS call if network coverage circumstances require this. These services are based on [PRD-IR.51] and [PRD-IR.92] for the voice call and [PRD-IR.94] for the video call.

Section 3.6 describes the architecture and technical enablers supporting 1-to-1 communication with Chatbots.

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 2.6.1.3) 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 2.11.

## 3.2 Messaging

### 3.2.1 1-to-1 Messaging Technology Selection

The technology selected for the first message in a 1-to-1 Conversation shall be based on:

- the RCS 1-to-1 Messaging technologies enabled i.e. RCS Standalone Messaging and/or RCS 1-to-1 Chat.
- the Chat capability of the contact (i.e. based on the last capability exchange that was executed according to the triggers defined in section 2.6.1)

The technology selection rules apply for the case where the originating client is registered for RCS services.

To a telephone number based address, i.e. a tel URI or SIP URI with a user=phone parameter, RCS 1-to-1 Chat is the preferred service, if enabled and based on the capability exchange it is supported by a contact, then RCS 1-to-1 Chat is used. Otherwise, if Standalone Messaging is enabled and available in the current coverage then Standalone Messaging is used. If neither of those can be used, xMS is used. Error scenarios are further detailed in section 3.2.1.1.

To a SIP URI that is not known to be linked to a Chatbot, that means

- a SIP URI without a user=phone parameter and
- where the domain part does not include the “botplatform” and
- where either

    - based on an earlier capability exchange or message exchange no indication was provided that the Contact is a Chatbot i.e. the +g.gsma.rcs.isbot feature tag defined in section 3.6.2.3 was not provided, or
    - an earlier messaging communication request was never referred to Chatbot communication as defined in section 3.2.1.1.

then RCS 1-to-1 Chat is the preferred service, if enabled and based on the capability exchange it is supported by a contact, then RCS 1-to-1 Chat is used. Otherwise, if Standalone Messaging is enabled and available in the current coverage then Standalone Messaging is used. Error scenarios are further detailed in section 3.2.1.1.

For the technology selected towards a Chatbot, see section 3.2.1.2.

#### 3.2.1.1 Messaging towards contacts not known as Chatbots

If the RCS 1-to-1 Chat Messaging technology is selected for a 1-to-1 conversation, the technology may change during the conversation based on the network fallback procedures defined in this document.

If the RCS 1-to-1 Chat Messaging technology and the initiation of a RCS 1-to-1 Chat session fails with the SIP 403 Forbidden response code including a warning header with warn-code set to “488" and the warn-text set to "Chatbot Conversation Needed”, then the client shall consider that the contact is a Chatbot and the technology selection described in section 3.2.1.2 shall be applied.

If the initiation of a RCS 1-to-1 Chat session fails with another SIP response code then the client shall

- send an RCS Standalone Message if RCS Standalone Messaging is enabled, otherwise
- send a SMS if the destination address is a telephone number based address.

If the RCS 1-to-1 Chat Messaging technology is selected for a 1-to-1 conversation and sending of a RCS 1-to-1 Chat message fails with one of the following MSRP response codes

- 400 request was unintelligible, or
- 403 attempted action is not allowed, or
- 501 recipient does not understand the request method, then

the client shall

- send an RCS Standalone Message if RCS Standalone Messaging is enabled, otherwise
- send a SMS if the destination address is a telephone number based address.

If the RCS Standalone Messaging service is selected for a 1-to-1 conversation, originating or terminating network fallback may be applied based on Service Provider policies.

If the RCS Standalone Messaging service is selected and sending an RCS Standalone Message to a destination address that is a telephone number based address returns SIP 403 Forbidden response code including a warning header with warn-code set to “488" and the warn-text set to "Chatbot Conversation Needed”, the client shall consider that the contact is a Chatbot and the message shall be resent applying the technology selection described in section 3.2.1.2

If the client receives one of the following responses:

- 380 Alternative Service, or
- 408 Request Timeout, or
- 486 Busy Here, or
- 487 Request Terminated, then

the client shall immediately resend the message as an xMS. For other error responses, the client will retry the message before falling back to xMS.

Table 15 provides an overview of the technology selection and fallback scenarios for 1-to-1 Messaging conversations.

| Enabled technologies | Terminating network chat fallback mechanism indicated | Non -RCS recipient | RCS recipient with Chat capability | RCS recipient with no Chat capability |
|---|---|---|---|---|
| **Only RCS Standalone Messaging** | N/A | RCS Standalone Messaging | RCS Standalone Messaging | RCS Standalone Messaging |
| **Only RCS 1-to-1 Chat Messaging** | Revocation supported | SMS | RCS 1-to-1 Chat Messaging unless latched to SMS | SMS |
| | Interworking supported | SMS | RCS 1-to-Chat Messaging | SMS |
| **Both RCS 1-to-1 Chat and RCS Standalone Messaging** | Revocation supported | RCS Standalone Messaging | RCS 1-to-1 Chat Messaging unless latched to SMS | RCS Standalone Messaging |
| | Interworking supported | RCS Standalone Messaging | RCS 1-to-Chat Messaging | RCS Standalone Messaging |

**Table 15: Messaging technology selection for 1-to-1 conversation initiation when A party is online**

#### 3.2.1.2 Messaging towards contacts known as Chatbots

An RCS client willing to establish a messaging communication with a SIP URI that is known to be linked to a Chatbot, as defined in section 3.6.2.4, shall select the messaging technology based on enabled technologies, i.e. through the value of the configuration parameter CHATBOT MSG TECH, as follows:

- For scenarios where capability discovery is enabled, Table 16 defines the technology that shall be used for Chatbot services.

| Enabled technologies | Chatbot with Capabilities for | | |
|---|---|---|---|
| | Chatbot Standalone Messaging only | 1-to-1 Chatbot Session only | Both 1-to-1 Chatbot session and Chatbot Standalone Messaging |
| Only Chatbot Standalone Messaging | Chatbot Standalone Messaging (see section 3.6.8.3.2) | Chatbot is not available | Chatbot Standalone Messaging (see section 3.6.8.3.2) |
| Only 1-to-1 Chatbot Session | Chatbot is not available | 1-to-1 Chatbot Session (see section 3.6.8.3.1) | 1-to-1 Chatbot Session (see section 3.6.8.3.1) |
| Both 1-to-1 Chatbot session and Chatbot Standalone Messaging | If anonymization does not apply Chatbot Standalone Messaging (see section 3.6.8.3.2), otherwise Chatbot is not available | 1-to-1 Chatbot Session (see section 3.6.8.3.1) | 1-to-1 Chatbot Session (see section 3.6.8.3.1) |

**Table 16: Messaging technology selection for Chatbot conversations**

- If capability discovery is not enabled, and if
  - only one of Chatbot Standalone Messaging or 1-to-1 Chatbot session is enabled, then the client shall select the enabled technology to establish the conversation with the Chatbot as defined in section 3.6.8.3. If a SIP 405 Method Not Allowed response is received with an Allow header indicating the methods understood by the Chatbot (i.e. the selected technology is not supported by the Chatbot), the Chatbot is not available and the client may resend the message through SMS if the conditions mentioned below are fulfilled.
  - both Chatbot Standalone Messaging and 1-to-1 Chatbot Sessions are enabled, then the client shall select to use a 1-to-1 Chatbot Session as defined in section 3.6.8.3.1. If a SIP 405 Method Not Allowed response is received including the MESSAGE method in the Allow header (i.e. the Chatbot supports only Chatbot Standalone Messaging), the client shall resend the message using Chatbot Standalone Messaging as defined in section 3.6.8.3.2.

If the RCS client is not registered when it is willing to establish a messaging communication or when the Chatbot services is not available in result of the technology selection procedure, then SMS may be used if the following conditions are fulfilled:

- An SMS address is known for the Chatbot through the Chatbot Information data (see section 3.6.4.1.3) and
- The user is sharing their MSISDN with the Chatbot (see section 3.6.5.1)
- The Chatbot is not known as a party sending spam (see section 3.6.6)
- The message to be sent is a regular text message i.e. not a File Transfer or one of the message types defined in section 3.6.10.1 and
- The RCS client is in coverage conditions where a SMS message can be sent.

### 3.2.2 Standalone messaging

#### 3.2.2.1 Overview

The technical realisation of the RCS Standalone Messaging is based on the OMA CPM Pager Mode and Large Message Mode mechanisms as described in [RCS-CPM-CONVFUNC-ENDORS].

The client is enabled for Standalone Messaging via the configuration parameter STANDALONE MSG AUTH defined in section A.1.3. If Standalone Messaging is enabled i.e. the configuration parameter STANDALONE MSG AUTH is set to "1" or "2", the client shall include the feature tags in the SIP REGISTER request as defined in section 2.4.4.1.

#### 3.2.2.2 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 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 [RCS-CPM-CONVFUNC-ENDORS].

The aggregation of IMDNs as specified in [RFC5438] shall not be used in RCS.

#### 3.2.2.3 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 shall, as specified in [RCS-CPM-CONVFUNC-ENDORS], push the deferred standalone messages once one of the clients of the target recipient RCS user becomes available.

If a deferred Standalone Message expires before it is delivered, the terminating Participating Function shall handle the deferred message by discarding it.

#### 3.2.2.4 Multidevice handling

RCS supports delivering of Standalone Messages to multiple devices. As described in [RCS-CPM-CONVFUNC-ENDORS], the delivery of 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 3.2.2.5.1.

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 [RCS-CPM-CONVFUNC-ENDORS].

#### 3.2.2.5 Standalone Message Delivery Assurance

##### 3.2.2.5.1 Interworking with Legacy Messaging services

Both the originating and the terminating network will offer interworking to xMS , where the former is used when sending a Standalone Message to a non-RCS user or a user of an operator with whom no interworking agreement for Standalone Messaging is offered. The latter option would be used towards RCS users whose primary device is offline, but reachable. This is considered to be providing the delivery assurance for Standalone Messaging except SIP MESSAGE request with the Chatbot IARI.

The [RCS-CPM-IW-ENDORS] document describes general interworking procedures applicable to both SMS and MMS and the realisation 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 [RCS-3GPP-SMSIW-ENDORS].

When handling an incoming SIP MESSAGE request with the Chatbot IARI described in section 3.6.2, the Messaging Server shall not perform interworking to xMS even if interworking is supported in general, since the incoming request includes the require and explicit parameters on a dedicated Accept-Contact header field carrying the Chatbot IARI feature tag defined in section 3.6.2.1.

In that case, the Chatbot platform can try to revoke the Chatbot messages as described in section 3.2.2.5.2 if required. Otherwise the SIP MESSAGE will be deferred in the Messaging Server as per section 3.6.9.

###### 3.2.2.5.1.1 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 [RCS-CPM-IW-ENDORS]. The actual interworking procedure is performed by the SMS and MMS gateways described in [RCS-3GPP-SMSIW-ENDORS] and [RCS-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.

###### 3.2.2.5.1.2 Interworking with SMS

When the target recipient device for an RCS Standalone Message is a non-RCS capable, SMS capable device, the process of interworking with legacy SMS is invoked according to [RCS5-CPM-CONVFUNC-ENDORS].

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 [RCS-CPM-IW-ENDORS] or [RCS-3GPP-SMSIW-ENDORS] respectively.

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.

#### 3.2.2.5.1.3 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 [RCS-CPM-CONVFUNC-ENDORS].

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 3.2.2.5.1.2 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 [RCS-CPM-IW-ENDORS].

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.

#### 3.2.2.5.2 Standalone Message Revocation

NOTE: Standalone Message Revocation is currently only used for SIP MESSAGE request with the Chatbot IARI described in section 3.6.2.

If required by the sender (i.e. currently only limited to the Chatbot Platform), the sender can generate MessageRevoke request as described in section 3.2.3.8.2.1. The MessageRevoke request is carried in the body of a SIP MESSAGE request that includes the same imdn.message-ID value of the Standalone Message that is intended to be revoked (as described in section 3.2.3.8.2.4). In this Standalone Message MessageRevoke SIP MESSAGE request, the following value shall be included instead of what required in section 3.2.3.8.2.1:

1. shall include an Accept-Contact header field with the CPM ICSI for Standalone Message, similarly to the case for IMDNs carried in SIP MESSAGE requests;
2. shall include a P-Preferred-Service header with the CPM ICSI for Standalone Message.

When the Messaging Server receives MessageRevoke requests, and Standalone Message Revocation is supported by it, the Messaging Server shall handle the request and shall send a MessageRevokeResponse request as described in section 3.2.3.8.2.2. The following value shall be included in the MessageRevokeResponse request instead of what required in section 3.2.3.8.2.2:

1. shall include an Accept-Contact header field with the CPM ICSI for Standalone Message, similarly to the case for IMDNs carried in SIP MESSAGE requests;
2. shall set the Request-URI of the MessageRevokeResponse request to the service-ids of the originating Chatbot defined as per section 2.5.4.1 that sent the message that is requested to be revoked;
3. shall include a P-Asserted-Service header with the CPM ICSI for Standalone Message;

### 3.2.2.6 Indicating of Capabilities in Pager Mode Messaging

This section defines a mechanism for the indication of client capabilities in a Pager Mode message. The sender client capabilities are conveyed in a SIP URI contained in the CPIM "From" header field of a Pager Mode Message.

The capability indication is added to the SIP URI using the "header" rule defined in section 25.1 of [RFC3261]. A "header" to convey capabilities is defined as follows:

```
capabilities = "Contact=" escaped-feature-param *("%3B" escaped-feature-
param)
     ; escaped-feature-param is feature-param defined in [RFC3840] with
     ; escaping of any characters not defined for use in [RFC3261] headers
```

This allows to include a Contact "header" in the URI the value of which shall be a list of feature tags as defined in [RFC3840] for which any characters in the feature tag name, value and separators that are not allowed to be used in a "header" according to [RFC3261] shall be escaped according to the rules defined in [RFC2396].

Non-normative examples:

- A CPIM "From" header of a Pager Mode message to convey the feature collection foo=1, bar=2:

  ```
  From: <sip:bob@example.com?Contact=foo%3D1%26bar%3D2>
  ```

- A CPIM "From" header of a Pager Mode message sent from the Chatbot Platform:

  ```
  From:
  <sip:anonymous@anonymous.invalid?Contact=+g.gsma.rcs.isbot%3B+g.gsma.
  rcs.botversion%3D%22%231%22>
  ```

#### 3.2.2.7 Large Messages via OMA CPM Standalone Messaging Pager Mode

##### 3.2.2.7.1 Procedures in the sending client

Rather than a fixed limit of 1300 bytes for switching over between CPM Pager Mode and Large Message Mode, for RCS the client shall select the Standalone Messaging mode based on the value of the configuration parameter STANDALONE SWITCHOVER SIZE defined in section A.1.3.

If the size of a Pager Mode message does not exceed the size provided in the configuration parameter value, then the OMA CPM Pager Mode mechanism shall be used, otherwise the client shall switch over to the OMA CPM Large Message Mode.

##### 3.2.2.7.2 Procedures in the originating Messaging Server

The procedures in the originating Messaging Server, if Standalone Messaging is used for interconnection with a terminating Messaging Server not supporting large pager mode messages, are based on bilateral agreement and are out of the scope of this document.

##### 3.2.2.7.3 Procedures in the terminating Messaging Server

Prior to delivery of a Pager Mode message to a client, the terminating Messaging Server shall check the size of the message.

If the size of the message does not exceed 1300 bytes, then terminating Messaging Server shall invoke the procedures for the delivery of the message via the OMA CPM Pager Message mode mechanism.

If the size of the message exceeds 1300 bytes and the Messaging Server did not receive the large pager mode feature tag defined in section 3.2.2.7.5 in the SIP REGISTER from the client, then terminating Messaging Server shall invoke the procedures for the delivery of the message via the OMA CPM Large Message mode mechanism.

If the size of the message exceeds a size of 1300 bytes and the Messaging Server did receive the large pager mode feature tag defined in section 3.2.2.7.5 in the SIP REGISTER from the client, then terminating Messaging Server shall invoke the procedures for the delivery of the message via the OMA CPM Pager Message mode mechanism.

#### 3.2.2.7.4 Procedures in the receiving client

The client receiving a Standalone Message via the OMA CPM Pager Mode mechanism shall accept messages regardless of the size of their message body, i.e. the reception shall not be authorized against a message size limit.

The client shall advertise this capability by adding the large pager mode support feature tag defined in section 3.2.2.7.5 in the Contact header of SIP REGISTER requests.

#### 3.2.2.7.5 Large Pager Mode support feature tag

The Large Pager Mode support feature tag is the indication of an OMA CPM client to support receiving a Standalone Message via the OMA CPM Pager Mode mechanism regardless of its size.

The indication of the large pager mode support is provided by a feature parameter as defined in section 9 of [RFC3840]. The feature parameter is encoded as an "other-tags" feature tag in accordance with the definitions of section 9 of [RFC3840].

The feature tag name shall be set to "g.gsma.rcs.cpm.pager-large". The media feature tag shall have no value.

Security considerations for this media feature tag are discussed in section 11.1 of [RFC3840].

The indication of the large pager mode support is represented as follows:

+g.gsma.rcs.cpm.pager-large

### 3.2.2.8 1-to-Many Messaging Technology Selection

1-to-Many Messaging builds on the 1-to-1 Messaging service and is based on OMA CPM procedures [RCS-CPM-CONVFUNC-ENDORS].

Technology selection rules are used to determine if OMA CPM based 1-to-Many Messaging is used or if it is provided using 1-to-1 Messaging per single recipient.

The availability of the 1-to-Many messaging service and the maximum number of participants allowed for 1-to-Many messaging services shall be controlled by the MAX 1 TO MANY RECIPIENTS parameter defined in sections A.1.3 and A.2.4. The MAX 1 TO MANY RECIPIENTS configuration parameter is applicable regardless of which technology is selected.

If the RCS Standalone Messaging service is enabled, the RCS Standalone Message service shall be selected based on procedures described in in sections 7.2.1 and 9.1 of [RCS-CPM-CONVFUNC-ENDORS] for sending a CPM Standalone Message to an ad-hoc group whereby the client shall set the copyControl attribute for all recipients to “BCC”.

If the RCS Standalone Messaging service is not enabled, the technology selected for the many 1-to-1 Messaging services is based on the 1 TO MANY SELECTED TECHNOLOGY client configuration parameter value (see sections A.1.3 and A.2.4). If the Chat service is selected via the 1 TO MANY SELECTED TECHNOLOGY configuration parameter, the procedures for technology selection and delivery assurance apply on a single recipient basis as defined for the RCS 1-to-1 Chat service.

### 3.2.3 1-to-1 Chat

#### 3.2.3.1 Overview

At a technical level, the Chat service 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*”.
- Delivery assurance information is included in SIP signalling by the network as per section 3.2.3.8.
- 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)

- For the messages sent in a Chat session, as well as the notifications mentioned above, the client shall request in the Instant Message Disposition Notification (IMDN) Common Profile for Instant Messaging (CPIM) header an Interworking Disposition Notification as defined in Appendix O of [RCS-CPM-CONVFUNC-ENDORS], if during the setup of the session the terminating network has indicated support for interworking using the corresponding feature tag defined in Table 17.
- When receiving an Interworking Disposition Notification as defined in Appendix O of [RCS-CPM-CONVFUNC-ENDORS], the client shall consider the message to have been delivered and interworked. The client shall thus not assume that a Delivery and a Display notification will follow.

- In normal circumstances, between two 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.
- 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), 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.
- Multimedia content within a Chat session is not permitted. 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 3.2.5 and 3.2.6.2) *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*. This also applies to requests generated by the Participating Function, and to responses generated by the Participating Function even if a response from the terminating client has not yet been received. To transfer multimedia content during a chat, File Transfer is used.
- When receiving a 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 upon detecting user activity, 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 and, consequently, all the diagrams shown in this document reflect this behaviour.
    b) The 200 OK response is sent immediately if the devices receiving the invitation are configured to auto-accept[13] the session invitations (IM SESSION AUTO ACCEPT configuration parameter defined in Table 85).

    Which behaviour to use, is configured via the IM SESSION AUTO ACCEPT configuration parameter defined in Table 85.

- 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. Based on the SIP signalling at session setup the client has access to the asserted identity of the other party as described in section 2.5.

[13] Note that the Service Provider multidevice policy has to be consistent with Chat auto-acceptance policy.

- The CPIM/IMDN wrapper shall be UTF-8 encoded to avoid any potential internationalisation issues.
- 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.
- The aggregation of IMDNs as specified in [RFC5438] shall not be used.
- 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 according to the rules and procedures of [RCS-CPM-CONVFUNC-ENDORS].
- The "IsComposing" notification is generated and processed according to the rules and procedures of [RCS-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 at protocol level be performed in a separate session. From the user experience perspective, they should be able to transfer files whilst in a Chat session. All multimedia content shall be transferred using File Transfer.

#### 3.2.3.2 Store and Forward Mode

Since Store and Forward is mandatory, all Messaging Servers shall support it. The Messaging Server serving the receiver of the message has the responsibility to store and forward messages which are not delivered. The Messaging Server serving the sender of the message has the responsibility of storing the delivered/displayed notifications if the sender of the message is offline.

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.

Stored messages and notifications are delivered to intended recipients as per the procedures defined in [RCS-CPM-CONVFUNC-ENDORS].

#### 3.2.3.3 Delivering stored disposition notifications

Deferred disposition notification shall be delivered as described in section 8.3.5 of [RCS-CPM-CONVFUNC-ENDORS].

To support backward compatibility with Messaging Servers based on earlier versions of this specification, a client shall support receiving the store and forward functionality using a special session for the purpose of delivering these notifications. As specified in section

7.3.11 of [RCS-CPM-CONVFUNC-ENDORS], this special session shall be automatically accepted by the device. It shall be 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.

#### 3.2.3.4 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 [RCS-CPM-CONVFUNC-ENDORS].
- The ISF described in [RCS-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 [RCS-3GPP-SMSIW-ENDORS] and [RCS-CPM-IW-ENDORS] which are responsible for doing the actual interworking (that is the protocol conversions) between RCS based chat and SMS or MMS.

When a Chat session invitation needs to be interworked, 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.

#### 3.2.3.5 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 Messaging Server forks the Chat session invitation to the different devices. Forking on the Messaging Server is further elaborated in [RCS-CPM-CONVFUNC-ENDORS].

NOTE: It is assumed that the originating user uses one device per session.

#### 3.2.3.6 Emoticons

Selected emoticons are displayed graphically but sent and received as text. The list of supported emoticons is defined in [RCS-CPM-CONVFUNC-ENDORS] Appendix L.

#### 3.2.3.7 Chat message size limitations

The maximum size of a text Chat message in bytes that a user can enter is controlled through the MAX SIZE IM configuration parameter defined in Table 85 in section A.1.3.

#### 3.2.3.8 1-to-1 Chat Delivery Assurance

##### 3.2.3.8.1 Network Fallback Support Capability

Delivery Assurance relies on a capability indication mechanism during the initiation of a 1-to-1 Chat session (see sections 3.2.3.8.3). The capability indication is used by the network to indicate which fallback mechanism shall be applied for messages sent by a client in this 1-to-1 Chat session (see section 3.2.3.8.3). The capability indication is used by the network to indicate its support for either

- the network fallback procedure, where the network is responsible for providing the fallback (also called Network Fallback to SMS, or NFS), or
- the message revocation, where the client is responsible for the fallback (also called Client Fallback to SMS or CFS).

The indication is provided by the feature tags defined in Table 17.

| Tag | Description |
|---|---|
| +g.gsma.rcs.msgrevoke | Message Revocation is supported (as defined in section 3.2.3.8.2) |
| +g.gsma.rcs.msgfallback | Network interworking is supported |

**Table 17: Feature tags used to indicate network support for chat fallback mechanisms**

Only one of the network indications defined in Table 17 can be present in a 1-to-1 Chat session, i.e. they are mutually exclusive.

The indication and the support of a network fallback mechanism is mandatory. This includes service provider deployments supporting termination of 1-to-1 Chat sessions only.

If the network indicates support of network fallback, the network shall take responsibility for delivering the message using the most suitable path. Therefore, if the terminating network has provided this indication, the client shall not apply procedures to monitor the delivery of chat messages and fallback.

When the CHAT REVOKE TIMER is set to a value higher than 0 and the network indicated capability is message revocation, the client shall apply procedures to monitor the delivery of chat messages and fallback.

When SMS latching is triggered, the client shall cache this event for future interactions with this contact unless Chat service is re-selected. This cached information shall affect the capability exchange triggers (see section 2.6) and the messaging technology selection (see section 3.2.1).

When the CHAT REVOKE TIMER parameter is set to 0 or the SMS fallback is disabled on the client and the network indicated capability is message revocation, then the client behaviour is based on the terminating network chat delivery policies (e.g. store and forward will be applied if the recipient cannot be reached).

#### 3.2.3.8.2 Chat Message Revocation

Message revocation is a feature that allows a client to request a chat message to be revoked by the recipient's Messaging Server. The recipient's Messaging Server processes MessageRevoke requests and responds with a MessageRevokeResponse request based on the chat message delivery status.

MessageRevoke requests and MessageRevokeResponse requests are not sent with CPIM headers, and a delivery and/or displayed notification shall not be requested.

There is no store and forward for the Message Revoke requests and the MessageRevokeResponse requests.

#### 3.2.3.8.2.1 Generating Chat MessageRevoke Requests

The MessageRevoke request is generated by the client of the message sender. 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 3.2.3.8.2.4).

MessageRevoke requests shall be generated only towards networks where their Messaging Server can handle them as described in section 3.2.3.8.2.3 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.

When message revocation is enabled (CHAT REVOKE TIMER, see section A.1.3), 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 3.2.3.8.2.3.

Message revocation is applicable for chat messages used for File Transfer via HTTP as defined in section 3.2.5 or Geolocation Push as defined in section 3.2.6 if an appropriate fallback mechanism is applicable for the message.

When a message is to be revoked, the client shall send a SIP MESSAGE request according to the rules and procedures of [RCS-CPM-CONVFUNC-ENDORS] with the clarifications listed here. In this SIP MESSAGE request, the client:

1. shall include an Accept-Contact header field with the CPM ICSI for Session Mode Messaging, similarly to the case for IMDNs carried in SIP MESSAGE requests;
2. shall add a dedicated Accept-Contact header field carrying the Message Revoke feature tag defined in section 3.2.3.8.2.3 along with the *require* and *explicit* parameters;
3. shall include the Content-Type header field with the value set to the message revocation content-type *application/vnd.gsma.rcsrevoke+xml*, as described in section 3.2.3.8.2.4;
4. shall include in the P-Preferred-Identity header the address of the originating RCS client that has been authenticated as per [RCS-CPM-CONVFUNC-ENDORS];
5. shall include a User-Agent header field as specified in Annex C.4.1;
6. shall set the Request-URI of the MessageRevoke request to the address of the target contact of the message that is requested to be revoked;
7. shall not include the device identifier of the original sender of the message in the MessageRevoke request;
8. shall set the body of the MessageRevoke request, as described in section 3.2.3.8.2.4, as follows:

a) The <Message-ID> element set to the value of the imdn.message-ID of the original message that is requested to be revoked,
b) The <From> element set to the URI of the sender of the message,
c) The <To> element set to the URI of the recipient of the message.

9. shall include a Conversation-ID header and a Contribution-ID header and shall

    a) If available, include the same Conversation-ID and Contribution-ID header field values that were used for the message being revoked; otherwise
    b) generate and include new Conversation-ID and Contribution-ID header field values;

10. shall include a P-Preferred-Service header with the CPM ICSI for Session Mode Messaging
11. shall send the SIP MESSAGE request according to the rules and procedures of [RCS-CPM-CONVFUNC-ENDORS].

#### 3.2.3.8.2.2 Handling MessageRevoke Requests

For a network, handling of MessageRevoke requests goes along with supporting clients that generate MessageRevoke requests.

The MessageRevokeResponse request shall indicate the result of the MessageRevoke request that can either have been successful or have failed. Similarly to the MessageRevoke request, the Messaging Server shall send a SIP MESSAGE request according to the rules and procedures of [RCS-CPM-CONVFUNC-ENDORS] with the clarifications listed here. In this SIP MESSAGE request, the Messaging Server handling the MessageRevoke request:

1. Shall include an Accept-Contact header field with the CPM ICSI for Session Mode Messaging, similarly to the case for IMDNs carried in SIP MESSAGE requests;
2. shall add a dedicated Accept-Contact header field carrying the Message Revoke feature tag defined in section 3.2.3.8.2.3 without the *require* and *explicit* parameters;
3. shall include the Content-Type header field with the value set to the message revocation content-type *application/vnd.gsma.rcsrevoke+xml* , as described in section 3.2.3.8.2.4;
4. shall include in the P-Asserted-Identity header the address of the intended recipient RCS Client, where the Messaging Server initiates the MessageRevokeResponse on behalf of the intended recipient that has been authenticated as per [RCS-CPM-CONVFUNC-ENDORS];
5. shall include a User-Agent header field of the Messaging Server as specified in Annex C.4.1;
6. shall set the Request-URI of the MessageRevokeResponse request to the address of the contact that sent the message that is requested to be revoked;
7. shall set the body of the MessageRevokeResponse request, as follows:

    a) The <Message-ID> element set to the value of the imdn.message-ID of the original message that is requested to be revoked,
    b) The <From> element set to the URI of the sender of the message,
    c) The <To> element set to the URI of the recipient of the message,

d) The <result> element set to the revoke result which shall be “success” if the MessageRevoke request was successful and “failure” if it has failed;

8. shall include a Conversation-ID header and a Contribution-ID header and shall
    a) if available, include the same Conversation-ID and Contribution-ID header field values that were used for the message being revoked; otherwise
    b) generate and include new Conversation-ID and Contribution-ID header field values;
9. shall include a P-Asserted-Service header with the CPM ICSI for Session Mode Messaging;
10. shall, when a 'tk' parameter set to ‘on’ is present in the P-Asserted-Identity of the corresponding MessageRevoke request, add a Privacy header as defined in [RFC3323] set to the value “tk”.
11. shall send the SIP MESSAGE request according to the rules and procedures of [RCS-CPM-CONVFUNC-ENDORS].

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 indicate the request as having 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.
- The message is under retry delivery attempt due to Messaging Server store and forward functionality.

The client shall ignore any MessageRevokeResponse request for chat messages that it does not recognize based on the Message-ID (corner case).

MessageRevoke requests shall never be forwarded to the client and shall be processed right after being received by the Messaging Server.

#### 3.2.3.8.2.3 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*.

#### 3.2.3.8.2.4 Message Revoke content-type

The Message Revoke XML schema is defined as shown on Table 18.

The associated Multipurpose Internet Mail Extensions (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 18: 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: 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 “failure”:

```
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: The Message-ID, "23499fuq34fu", in the XML body refers to the CPIM Message-ID of the message that was revoked.

#### 3.2.3.8.3 Chat Fallback Mechanism management in the Client

When initiating a 1-to-1 Chat session, the client shall monitor the delivery of the messages exchanged in the session based on the feature tags defined in Table 17:

- If a SIP 200 OK response is received as final response, the client shall behave as follows based on the presence of the feature tags defined in section 3.2.3.8.1.
  - If the Contact header in the 200 OK response included the message revocation feature tag defined in section 3.2.3.8.1 while the CHAT REVOKE TIMER client configuration parameter is configured with the value 0 or if applicable the SMS fall-back is disabled on the client, the client shall assume delivery for any messages based on the terminating network delivery policies (e.g. store and forward will be applied if the recipient cannot be reached).
  - If the Contact header in the 200 OK response includes the message revocation feature tag defined in section 3.2.3.8.1, the CHAT REVOKE TIMER client configuration parameter is set to a value higher than 0 and if applicable the SMS

fall-back is enabled on the client, the client shall apply monitoring of the delivery for any messages sent within the session (i.e. through a timer based on the CHAT REVOKE TIMER client configuration parameter).

- If the Contact header in the 200 OK response includes the interworking feature tag defined in section 3.2.3.8.1, the client shall not monitor the delivery of any messages that it sends in the Chat session.

For a client invited to a 1-to-1 Chat session (excluding a session for delivering stored messages or notifications), behaviour shall be as follows:

- If the Contact header field included in the SIP INVITE request included the message revocation feature tag defined in section 3.2.3.8.1 while the CHAT REVOKE TIMER client configuration parameter is configured with the value 0 or if applicable SMS fall-back is disabled on the client, the client shall assume the delivery of any messages that it sends in the Chat session according to the terminating network delivery policies (e.g. store and forward will be applied if the recipient cannot be reached).
- If the Contact header in the SIP INVITE request includes the message revocation feature tag defined in section 3.2.3.8.1 and the CHAT REVOKE TIMER client configuration parameter is set to a value higher than 0 and if applicable SMS fall-back is enabled on the client, the client shall monitor the delivery of any messages that it sends in the Chat session (i.e. through a timer based on the CHAT REVOKE TIMER client configuration parameter).
- If the Contact header of the SIP INVITE request includes the interworking feature tag defined in section 3.2.3.8.1, the client shall not monitor the delivery of the messages that it sends in the Chat session.

#### 3.2.3.8.4 Chat Fallback Mechanism Management in the network

Service Providers supporting the RCS 1-to-1 Chat service shall support network fallback. The fallback is performed in the network serving the recipient of the message. RCS 1-to-1 Chat Fallback in the network serving the sender of the message is out of the scope of this document. The following procedures shall be implemented:

- When handling an SIP INVITE request for a 1-to-1 Chat session, the messaging server in the originating network shall add either the message revocation or the network interworking feature tag defined in section 3.2.3.8.1 in the Contact header field of the SIP INVITE request sent towards the terminating client based on the chat fallback mechanism that is supported.
- When handling an SIP INVITE request for a 1-to-1 Chat session, the messaging server in the terminating network shall add either the message revocation or the network interworking feature tag defined in section 3.2.3.8.1 in the Contact header field in every 200 OK to the SIP INVITE request sent towards the originating client.

NOTE: For networks with RCS Standalone messaging service enabled that enable only receiving 1-to-1 chat messages (i.e. CHAT AUTH is set to 0 and GROUP CHAT AUTH is set to 1) and that choose to interwork with Service Providers that enable only RCS 1-to-1 Chat service (option 2) by deploying an RCS Standalone message to RCS 1-to-1 Chat session interworking function, the above procedure can be fulfilled by the interworking function.

The terminating network shall always return a SIP 200 OK response to a SIP INVITE request for 1-to-1 chat session since the network takes responsibility for delivering the message in the best way possible regardless of the connectivity status of the client.

If the terminating network supports Message Revocation, this shall be implemented as defined in this document.

#### 3.2.3.8.5 Network indicates delivery as SMS

When a network supports interworking as per section 3.2.3.4 and it delivers the message through interworking to SMS, the network shall generate an interworking notification as defined in Appendix O of [RCS-CPM-CONVFUNC-ENDORS] when such a notification was requested in the IMDN Disposition-Notification CPIM header of the message that was interworked. In that case, the network shall not generate a Delivery disposition notification. If IMDN Disposition-Notification CPIM header did not include a request for an interworking notification, the network shall generate a Delivery Notification instead.

NOTE: Since a client will always have requested an interworking notification, this requirement to generate a regular Delivery notification is intended to support clients of previous RCS versions.

#### 3.2.3.8.6 Procedures for Client SMS Fall-back

If according to the procedures defined in section 3.2.3.8.1 the client runs a timer based on the CHAT REVOKE TIMER client configuration parameter for messages in a conversation, and the client receives "delivery" disposition notifications for all messages in the conversation, then the client shall stop the timer.

The client shall start a timer based on the CHAT REVOKE TIMER client configuration parameter for the next message sent in the conversation considering the message revocation capability indicated in accordance with the procedures defined in section 3.2.3.8.1.

1. If the timer based on the CHAT REVOKE TIMER client configuration parameter is running for at least one conversation and if

    a) the client registers in IMS successfully due to a previous de-registration (e.g. due to user setting or data-off) and if the value of the RECONNECT GUARD TIMER configuration parameter defined in sections A.1.3 and A.2.4 is not set to "0", then the client shall start the reconnection guard timer with the value provided in the RECONNECT GUARD TIMER configuration parameter defined in sections A.1.3 and A.2.4, or
    b) the client re-connects to the Proxy Call Session Control Function (P-CSCF) due to a previous loss of connection to the P-CSCF and if the value of the RECONNECT GUARD TIMER configuration parameter defined in sections A.1.3 and A.2.4 is not set to "0", then the client shall start the reconnection guard timer with the value provided in the RECONNECT GUARD TIMER configuration parameter defined in sections A.1.3 and A.2.4.
    c) The reconnect guard timer shall start if

i. a success response is received from the network for an initial registration or re-registration resulting from the client procedures to re-connect to the P-CSCF, otherwise
ii. at the time of P-CSCF connection re-gain.

2. If the timer based on the CHAT REVOKE TIMER client configuration parameter expires, then processing commences with step 3.
3. The client shall check whether the reconnection guard timer is running. If yes, then processing commences with step 5, otherwise processing commences with step 4.
4. If the client

a) is not registered in IMS due to missing data connection (e.g. data off, loss of connection to the P-CSCF and registration expired), then the client shall wait until data connection is regained.
b) if the data connection is regained then the client shall send an initial registration as per procedures of section 2.4.
c) has stored a valid IMS registration but it has previously detected a loss of connection to the P-CSCF, then the client shall wait until data connection is regained.
If the data connection is regained

i. and the IMS registration is valid (e.g. registration is not expired, the client IP address did not change, access network did not change), then the client shall send a re-registration as per procedures of section 2.4.
ii. and the IMS registration is not valid (e.g. registration is expired, the client IP address did change), then the client shall send an initial registration as per procedures of section 2.4.

If the initial registration or the re-registration is successful, then

iii. if the value of the RECONNECT GUARD TIMER configuration parameter defined in section A.1.3 and A.2.4 is set to "0" then processing commences with step 6.
iv. otherwise, the client shall start the reconnection guard timer with the value provided in the RECONNECT GUARD TIMER configuration parameter defined in sections A.1.3 and A.2.4. Processing commences with step 5.

d) otherwise, in all other cases where the client is registered in IMS, processing commences in step 6.

5. If the reconnection guard timer is running and

a) if "delivery" disposition notifications have been received for all messages in the conversation, then the reconnection guard timer shall be stopped by the client. The client shall stop the message fall-back processing.
b) if the client detects a loss of connection to the P-CSCF or it is de-registered, the reconnection guard timer is stopped and processing commences with step 4

c) If the reconnection guard timer expires, the processing commences with step 6.

6. The client shall verify whether connectivity for sending SMS messages exists. If connectivity for sending of SMS messages exists, then processing commences with step 7. Otherwise, the client

   a) shall wait until connectivity for SMS is regained again.
   b) If the connection to the P-CSCF is lost or the client is de-registered from IMS while waiting for connectivity for sending of SMS messages, the client shall stop waiting for SMS connectivity and continue processing in step 4.
   c) If "delivery" disposition notifications have been received for all messages in the conversation, then the client shall stop waiting for SMS connectivity. The client shall stop the SMS fall-back processing.
   d) Once connectivity for sending SMS is regained, then processing commences with step 7.

7. If a user authorisation is required for the SMS fall-back for this conversation, then the client shall invoke a user interaction. Otherwise processing commences with step 8.

   a) If the client receives "delivery" disposition notifications for all messages in the conversation, then the client shall abort the user interaction and stop the message fall-back processing.
   b) If the user interaction results in the user authorisation of the SMS fall-back, then the procedure commences with step 4. The client shall retain the user authorisation of the SMS fall-back for the subsequent processing, i.e. the client shall not invoke the user interaction again.
   c) If the user interaction results in rejection of the message fall-back, then the client shall stop message fall-back processing.

8. The client shall create a Message Revoke request for the oldest chat message of the conversation for which no "delivery" disposition notification has been received and which has not been processed for SMS fall-back. The client shall send the Message Revoke request to the network. Message revocation shall be implemented as defined in section 3.2.3.8.2. If the value of the configuration parameter CFS TRIGGER defined in sections A.1.3 and A.2.4

   a) is set to "1", then

      i. if the Message Revocation request fails due to loss of connection to the P-CSCF, then the client shall continue processing with step 4.
      ii. if the client receives a success response (200 OK) to the Message Revoke request, then the client shall

   1. start an operation timer to supervise the processing of the Message Revocation, otherwise
   2. not send a fall-back SMS and consider the chat message as processed for SMS fall-back. The client shall continue with the procedure in step 8 until all messages in the conversation have been processed.

      iii. If the client receives a "delivery" disposition notification for the message to be revoked, it shall stop SMS fall-back handling.

iv. If the client receives a Message Revocation response from the network with a "success" result, then the client shall stop the operation timer and shall send the fall-back SMS following the procedures for fall-back according to the type of message (e.g. chat message, file transfer).
v. If the client receives a Message Revocation response from the network with a "failure" result, then the client shall stop the operation timer and shall not send the fall-back SMS and consider the chat message as processed for SMS fall-back. The client shall continue with the procedure in step 8 until all messages in the conversation have been processed.
vi. If the client detects a loss of connection to the P-CSCF or the client is de-registered from IMS (e.g. due to user settings), then the client shall stop the operation timer and shall send the fall-back SMS following the procedures for fall-back according to the type of message (e.g. chat message, file transfer). If there is at least one message in the conversation for which no "delivery" notification has been received and which has not been processed for SMS fall-back the client shall continue processing with step 4, otherwise is shall stop SMS fall-back handling.
vii. If the operation timer expires then the client shall send the fall-back SMS following the procedures for fall-back according to the type of message (e.g. chat message, file transfer).
viii. If submission of the SMS to the network fails (e.g. no SMS connectivity), then the client shall suspend the processing of SMS fall-back and apply the client procedures for the handling of failed SMS message submissions.
ix. If the submission of the SMS is confirmed by the network with a success response, then the client shall consider the chat message as processed for SMS fall-back. The client shall trigger SMS latching and cache this status for the contact. The client shall continue with the procedure in step 8 until all messages in the conversation have been processed.
x. If the submission of the SMS is rejected by the network with a failure response, then the client shall stop processing of the SMS fall-back and apply the client procedures for the handling rejected SMS message submissions

b) is set to "0" or is not present then the client shall send the fall-back SMS following the procedures for fall-back according to the type of message (e.g. chat message, file transfer) and the Message Revoke request at the same time.

i. If the Message Revoke request fails due to loss of connection to the P-CSCF and there is at least one message in the conversation for which no "delivery" notification has been received and which has not been processed for SMS fall-back, then the client shall continue processing with step 4, otherwise the client shall stop processing of SMS fall-back.

ii. If submission of the SMS to the network fails (e.g. no SMS connectivity), then the client shall suspend the processing of SMS fall-back and apply the client procedures for the handling failed SMS message submissions.
iii. If the submission of the SMS is confirmed by the network with a success response, then the client shall consider the chat message as processed for SMS fall-back. The client shall trigger SMS latching and cache this status for the contact. The client shall continue with the procedure in step 8 until all messages in the conversation have been processed.
iv. If the submission of the SMS is rejected by the network with a failure response, then the client shall stop processing of the SMS fall-back and apply the client procedures for the handling rejected SMS message submissions.
v. The outcome of the Message Revocation operation does not alter the processing requirements of the client with regard to SMS fall-back. However, the client may need to inspect the outcome of the Message Revocation operation to satisfy the requirements for the presentation of the message status to the user.

#### 3.2.3.9 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 3.2.3.1 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 order to interleave the incoming and outgoing messages. Please note that the shown message time at the UX should be based on the network time (i.e. the CPIM DateTime header, 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 synchronised with the Message Store Server as specified in section 4.1.

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.).

### 3.2.4 Group Chat

#### 3.2.4.1 Overview

The technical realisation of Group Chat is based on the CPM Ad-hoc Group and the CPM Long-Lived Group Session as defined in [RCS-CPM-CONVFUNC-ENDORS] with the additional requirements and clarifications defined in this document.

The support of CPM Long-Lived Group Session is mandatory for the Messaging Server. This requires the conference focus of the Messaging Server to keep the Group Chat participant and meta-information beyond an active Group Chat session. The conference focus shall keep the participant information and meta-information for idle Group Chats for a minimum time as defined in [PRD-RCC.71].

The support of CPM Long-Lived Group Session is mandatory for the client. The value of the Conversation-ID header defined in [RCS-CPM-CONVFUNC-ENDORS] provides the unique ID of an individual RCS Group Chat. It shall be used by the client to associate Group Chat sessions, Chat messages, participant and meta-information with the corresponding Group Chat conversation.

#### 3.2.4.2 Authorisation of Group Chat

The client is authorised to offer the Group Chat service to the user if the configuration parameter GROUP CHAT AUTH as defined in section A.1.3 is set to "1".

If the client is not authorised for Group Chat and it receives an invitation for a Group Chat session, then it shall reject it with a SIP 488 NOT ACCEPTABLE HERE response.

#### 3.2.4.3 Initiation of the Group Chat

The client shall offer the user to initiate a Group Chat only with contacts being capable of the Chat service as defined in section 2.6.1.3.

NOTE: The caching of addresses as defined in section 2.5.2.1 and 2.5.3.1 in combination with the fact that the Chat capability of the contact must be known will result in the client always providing the addresses of invited participants in international format.

For the establishment of a Group Chat the client shall request the user to select at least 2 contacts capable of the Chat service. The client shall allow the user to select a number of Chat service capable contacts up to the limit defined in the configuration parameter MAX_AD-HOC_GROUP_SIZE, reduced by one (the initiator itself), as defined in section A.1.3.

Extending of a 1-to-1 Chat to a Group Chat as defined in [RCS-CPM-CONVFUNC-ENDORS] is not applicable for this version of RCS.

The client shall allow the user to assign Group Chat meta-information such as subject and an icon to the Group Chat. The subject shall be conveyed to the invitees during Group Session establishment as defined in [RCS-CPM-CONVFUNC-ENDORS]. The icon shall be conveyed to the invitees via the procedures for managing and receiving Group Chat meta-information described in section 3.2.4.7 and 3.2.4.8 respectively.

Prior to initiation the client shall assign to the Group Chat a unique value for the Contribution-ID and Conversation-ID headers defined in [RCS-CPM-CONVFUNC-ENDORS]. Both Contribution-ID and Conversation-ID headers shall be set to the same value.

The client shall derive the Request-URI of the SIP INVITE from the configuration parameter CONF-FCTY-URI defined in section A.1.3.

The client shall initiate a Group Chat following the definitions of [RCS-CPM-CONVFUNC-ENDORS] for the initiation of a CPM Group Session for a CPM Ad-hoc Group.

Once a 200 OK response is received for the Group Chat invitation, the client shall store the values of the Conversation-ID and Contribution-ID SIP headers and the Group Chat Session Identity from the Contact header along with the subject of the Group Chat.

#### 3.2.4.4 Invitation to a Group Chat

A client is invited to a new Group Chat if it receives a session invitation for a CPM Group Session for a CPM Ad-hoc group as defined in [RCS-CPM-CONVFUNC-ENDORS] with a value of the Conversation-ID header which is not known by the client. The client shall accept the invitation to the Group Chat in accordance with the value of the configuration parameter IM SESSION AUTO ACCEPT GROUP CHAT as defined in section A.1.3.

Once the invitation to the Group Chat is accepted, the client shall store the values of the Contribution-ID and Conversation-ID SIP headers and the Group Chat Session Identity from the Contact header along with the subject of the Group Chat.

#### 3.2.4.5 Content type negotiation

When establishing CPM Group Sessions, the clients and the entities of the Messaging Server shall negotiate the supported content types for the CPM Group Session as defined in [RCS-CPM-CONVFUNC-ENDORS] with the following additional clarifications and requirements:

- In the SDP of the SIP INVITE request and response, the a=accept-types attribute shall include *message/cpim*, and the content-type for the conference-info object if supported.
- 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 is supported (see section 3.2.5) then the a=accept-wrapped-types attribute shall also include *application/vnd.gsma.rcs-ft-http+xml*.
- If Geolocation PUSH is supported (see section 3.2.6.2), then the a=accept-wrapped-types attribute shall also include *application/vnd.gsma.rcspushlocation+xml*.
- To transfer any other content (e.g. multimedia content) during a chat, File Transfer is used.
- The client and the Messaging Server shall take the procedures defined in section 3.2.5.2 for File Transfer into account.
- The client and the Messaging Server shall take the procedures defined in section 3.2.6.2 for Geolocation PUSH into account.

#### 3.2.4.6 Adding participants to a Group Chat

The client shall allow the user to add additional participants to a Group Chat, if they are capable to support Chat and if they are not member of the Group Chat already.

NOTE: The caching of addresses as defined in section 2.5.2.1 and 2.5.3.1 in combination with the fact that the Chat capability of the contact must be

known will result in the client always providing the addresses of invited participants in international format.

This includes participants listed in the most recent conference-info document received by the client if their endpoint status is set to "disconnected" with disconnection method set to "departed" or "failed".

The client shall allow the user to add more participants to a Group Chat if the value of the maximum-user-count element is greater than the value of the user-count element of the latest conference-info document defined in [RFC4575] and received by the client as per procedures defined in section 3.2.4.8. The client shall allow the user to only add a number of additional contacts that does not exceed the value of the maximum-user-count element.

Based on the client procedure above it is mandatory for the Messaging Server to provide the values of maximum-user-count and the user-count elements in all conference-info documents.

The client shall add new participants to a Group Chat in accordance with the procedures for the inviting other participants to a CPM Group Session as defined in [RCS-CPM-CONVFUNC-ENDORS].

#### 3.2.4.7 Managing Group Chat meta-information

Group Chat meta-information comprises additional characteristics of a Group Chat which can be managed by the participants.

The client and the Messaging Server shall support the procedures for the management of

- the subject
- the icon, and
- user roles

via the procedures defined in [RCS-CPM-CONVFUNC-ENDORS] for the Group Session Data Management.

For the request to change the icon, the client shall use the request operation to set the icon using the <file-info> element of the <file> element as defined in [RCS-CPM-CONVFUNC-ENDORS]. The use of the <icon-uri> to set the icon is not applicable for this version of RCS.

In addition, the client and the Messaging Server shall support the procedures related to the indication of the Group Chat policy as defined in [RCS-CPM-CONVFUNC-ENDORS].

If for a Group Chat session, the Group Chat Session Data Management is not available as defined in [RCS-CPM-CONVFUNC-ENDORS], then

- the client shall store the value of the subject assigned at the time of initiation of the Group Chat persistently. The client shall use the value of the subject in SIP INVITE and SIP REFER requests to restart a Group Chat or when adding participants to a Group Chat respectively. The client shall ignore the value of the subject header when receiving invitations for the stored CPM Group Session.
- the client shall not offer functions related to the management of icon and user roles to the user.

The procedures for the client and the Messaging Server related to the user experience of user roles and the group policies are defined in [PRD-RCC.71].

#### 3.2.4.8 Receiving Group Chat participant and meta-information

The client shall determine the Group Chat participant information (i.e. the list of participants) and Group Chat meta-information by the following means:

- During an active Group Chat session via

  - the conference-info document via MSRP SEND requests sent by the network automatically at the time of session set-up or after participant information or meta information has changed, or
  - via SIP NOTIFY as a result of the client's subscription to the conference event package

  as defined in [RCS-CPM-CONVFUNC-ENDORS].
  The policy for the client to receive Group Chat participant and meta-information via MSRP is determined by the network based on the negotiation of the content-type for the conference-info object as defined in [RCS-CPM-CONVFUNC-ENDORS]. If the network supports delivery of Group Chat participant and meta-information via MSRP, then the client shall not subscribe to the conference event package. Otherwise, if the network does not support delivery of Group Chat participant and meta-information via MSRP, then the client shall subscribe to the conference event package and in this case, the network will not send any meta-information via MSRP.

- Outside of an active Group Chat session, if enabled, via the Group State Objects or Conference Information Objects stored in the Common Message Store using the procedures defined in section 4.1.

The client shall store the latest list of participant addresses locally.

If the client receives an update of participant or meta-information, then it shall compare the received list of participants or the meta-information event with the stored list of participants and meta-information and shall notify the user about changes in the participant list or meta-information only as a result of the comparison.

For the reception of icons assigned to the Group Chat, the client shall support both the <icon-uri> and the <file-info> element of the conference-info document extension defined [RCS-CPM-CONVFUNC-ENDORS].

To download the icon file using the URL contained in the <icon-uri> element, the client shall invoke the file download procedure as defined in section 3.2.5.3.2.1 for download of files in a Group Chat. If the configuration parameter FT HTTP DL URI defined in section A.1.4 is present, then the client shall follow the procedure as defined for this case in section 3.2.5.3.2.1 taking the following requirement into account. When creating the download URL, the client shall not add the "id" parameter to the download URL.

#### 3.2.4.9 Closing Group Chat sessions

##### 3.2.4.9.1 Group Chat session idle time

The conference focus of the Messaging Server shall take responsibility to close idle Group Chat sessions. In this case, the participants of the Group Chat remain members of the Group Chat. A participant can restart the Group Chat session at any time using the procedure for the restart of a Group Chat. The conference focus of the Messaging Server shall end the Group Chat session using the procedure for the CPM Group Chat session ending request for session inactivity defined in [RCS-CPM-CONVFUNC-ENDORS]. The maximum allowed idle time in the conference focus shall not be larger than 300 seconds. When measuring the idle time in the conference focus, MSRP messages for disposition notifications shall not be taken into account.

A Messaging Server not acting as the conference focus may also monitor Group Chat sessions for becoming idle. Such a Messaging Servers shall apply a significantly larger idle timer value than the timer value defined for the conference focus.

The client can leave the Group Chat session involuntarily, e.g. at the time to shut down or power off. In this case, the client or an entity in the network detecting the event shall generate a SIP BYE request including a Reason Header field set to SIP and the protocol-cause set to a value other than 200. 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") in accordance with the definitions in [3GPP TS 24.229] for bearer loss detected by the P-CSCF.

##### 3.2.4.9.2 Group Chat termination Service Provider policies

The conference focus of the Messaging Server may close a Group Chat permanently based on Service Provider policy if

- less than the minimum number of participants is reached or
- the Group Chat has not been used by the participants for a time defined by the Service Provider.

If a Service Provider Policy applies and the condition for Group Chat permanent termination is met, then the conference focus of the Messaging Server shall remove all participants of the Group Chat using the procedure for the CPM Group Chat session ending request defined in [RCS-CPM-CONVFUNC-ENDORS]. If the client receives the CPM Group Chat session ending request, then it shall apply the UX procedures for a Group Chat where the user is no longer participating.

#### 3.2.4.10 Restart of the Group Chat session

If a Group Chat session was closed due to inactivity and a client requires an active Group Chat session for the processing of user requests, then the client shall restart the Group Chat session using the procedure defined in [RCS-CPM-CONVFUNC-ENDORS] for re-joining a CPM Group Chat session. The client shall use the stored values of the Group Chat Session Identity, Contribution-ID and Conversation-ID to re-join the Group Chat session.

The Group Chat is restarted if the client receives a 200 OK response in accordance with the procedures defined in [RCS-CPM-CONVFUNC-ENDORS].

If the SIP INVITE for the restart of the Group Chat session fails with a SIP 404 Not Found response in accordance with the procedures in [RCS-CPM-CONVFUNC-ENDORS], the Service Provider closed the Group Chat based on Service Provider policy in accordance with the definitions in section 3.2.4.9.2. The client shall apply the UX procedures for a Group Chat where the user is no longer participating.

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 Group Chat Session Identity) (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 section 3.2.4.15), 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 Back-to-Back User Agent (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 Group Chat Session Identity) 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").

If the SIP INVITE for the restart of the Group Chat session fails with a SIP 403 Declined response including a warning header set to "127 Service not authorized" in accordance with the procedures in [RCS-CPM-CONVFUNC-ENDORS], then the user is not authorised to restart the Group Chat. The client shall apply the user interface procedures for a Group Chat where the user is no longer participating.

If a SIP response other than the ones defined in [RCS-CPM-CONVFUNC-ENDORS] is received by the client, then the client shall not alter the status of the Group Chat. The client may retry the Group Chat restart.

A client receiving an invitation to a Group Chat session with a known Conversation-ID shall accept the request automatically, unless there is a pending user request to leave the Group Chat.

#### 3.2.4.11 Leaving a Group Chat

If the user requests the client to leave a Group Chat, and a Group Chat session exists, then the client shall send a SIP BYE request following the procedure defined in [RCS-CPM-CONVFUNC-ENDORS] for leaving a Group Chat Session.

If the user requests the client to leave a Group Chat, and a Group Chat session does not exist, the client shall restart the Group Chat session first as described in section 3.2.4.10, and then follow the procedure defined in [RCS-CPM-CONVFUNC-ENDORS] for leaving a Group Chat Session.

The client may also leave a Group Chat at the time Group Chat session restart by rejecting the Group Chat invitation with a 603 Decline response.

#### 3.2.4.12 Removal of a participant

The client and the Messaging Server shall support the procedures for a participant to remove another participant as defined [RCS-CPM-CONVFUNC-ENDORS]. The client shall offer the user removal of a participant only if it is entitled by the participant removal policy

and the user's own role as indicated by the Messaging Server. If the participant removal policy is absent in the Group Chat, the client shall not offer the user the possibility to remove other participants.

#### 3.2.4.13 Procedures in the Messaging Server for the removal of a user's IMS identity and profile

If the Service Provider removes the IMS identity and the profile of a user, then the Messaging Server shall invoke the procedures defined in [RCS-CPM-CONVFUNC-ENDORS] for the termination of all CPM Sessions due to administrative reasons.

In addition to the procedures defined in [RCS-CPM-CONVFUNC-ENDORS] for the creation of a CPM Group Session with a participant the conference focus shall, if it receives a SIP 404 Not Found response to the SIP INVITE, remove the corresponding participant from the participant list of the Group Chat. The conference focus shall inform the other participants via an update of the participant information using the procedures described in section 3.2.4.8. The elements of the removed participant's user endpoint in the conference-info document shall be set as follows:

- Status of the user endpoint element is set to "*disconnected*",
- Disconnection-method of the user endpoint element is set to "*departed*".

#### 3.2.4.14 Delivery and Display notifications

The client and the Messaging Server shall apply the procedures for Delivery and Display notifications as defined in [RCS-CPM-CONVFUNC-ENDORS] for the handling of disposition notifications for Group Chat.

The additional requirements for the composition of disposition notifications defined in section 3.2.4.17 shall apply also, if disposition notifications are sent via SIP MESSAGE.

#### 3.2.4.15 Support of Store and Forward

The Messaging Server and the client shall support Store and Forward of Chat messages, and Group Chat participant and meta information via the procedures defined in [RCS-CPM-CONVFUNC-ENDORS] for deferred CPM Group Session delivery.

#### 3.2.4.16 Multidevice handling and the Common Message Store

If Multidevice and the Common Message Store is supported by the Service Provider the Messaging Server shall support forking of Group Chat sessions to multiple clients as defined in in [RCS-CPM-CONVFUNC-ENDORS] for CPM Group Session establishment. The Messaging Server shall act as a Back-to-Back User Agent (B2BUA) in this case. Delivery of sessions and messages to non-CPM devices as defined in [RCS-CPM-CONVFUNC-ENDORS] is not applicable for Group Chat.

Multiple clients of a user will receive the Group Chat participant and meta-information during a group Chat session as described in section 3.2.4.8.

In addition, based on Service Provider Policy the Messaging Server may support direct delivery of messages in a CPM Group Session as defined in in [RCS-CPM-CONVFUNC-ENDORS]. For the client the support of direct delivery of messages is mandatory.

If the Common Message Store is deployed by the Service Provider then the client shall retrieve the conversation history including participant and meta-information using the procedures described in section 4.1.

#### 3.2.4.17 Media plane handling

In addition to the definitions in [RCS-CPM-CONVFUNC-ENDORS], client and servers shall apply the following procedures for media plane handling in a Group Chat Session.

To prevent revealing the user identity when transmitted over unprotected links, the client should set the value of the CPIM To header for Chat messages in a Group Chat to *sip:anonymous@anonymous.invalid*. The client shall set the value of the CPIM From header to its public user identity and include the user's display name.

For Delivery and Display notifications, the client shall set the value of the CPIM To header to the identity of the original sender of the message, taken from the value of the CPIM From header of the chat message it relates to.

The originating Messaging Server shall always set the value of 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.

The maximum size of a text Chat message in bytes that a user can enter in the Group Chat is controlled via the configuration parameter MAX SIZE IM as defined in section A.1.3.

#### 3.2.4.18 Group Chat Session Identity

The Group Chat Session Identity is conveyed during the establishment of SIP session as the value of the Contact header. Intermediate nodes in the path between the Messaging Server and the client shall transparently forward the contact URI if the Contact header field contains the “isfocus” feature tag. For IMS entities, this behaviour is applied in accordance with the definitions in [3GPP TS 24.229]. For CPM entities this behaviour is applied in accordance with the definitions in [RCS-CPM-CONVFUNC-ENDORS].

#### 3.2.4.19 Connection Model for Subscriptions from Participating Function to Controlling Function

The Messaging Server Participating Function acts as a subscriber to the conference event package of a Group Chat controlled by the Messaging Server Controlling Function. During an active subscription, the Messaging Server Controlling Function notifies the Participating function about changes of the Group State status and participant list.

Messaging Server Participating Functions and Controlling Function may reside in different Service Provider networks. The Connection Model applied on the interface between the two functions shall minimise the impact on the Controlling Function coming from the Participating Function Service Provider's network topology and service provisioning. This is achieved by limiting the number of active subscriptions per participant in a given Group Chat to "1" based on the Connection Model below.

If the Participating Function initiates a new subscription for a Group Chat Session Identity on behalf of a participant, the Controlling Function shall accept the request provided that the Group Chat Session Identity exists and the participant is authorised to subscribe to it, i.e. it is a participant in the Group Chat.

At the time of acceptance, if the Controlling Function has another subscription active for the same focus session identity and participant combination then the older subscription should be terminated as defined in [RFC6665].

The Service Provider of the Participating Function shall ensure that the connection model towards the Controlling Function does not restrict the service provided to its users. For Service Providers offering Group Chat for Multidevice the application of a Back-to-Back User Agent (B2BUA) for subscriptions in the Messaging Server Participating Function is mandatory.

### 3.2.5 File Transfer

#### 3.2.5.1 Overview

This section describes the File Transfer mechanism that is based on

- the originating client storing the file in the HTTP Content Server,
- use of Standalone Messaging, 1-to-1 Chat and Group Chat procedures described in sections 3.2.2, 3.2.3 and 3.2.4 to transport the file location URI to the recipient(s),
- the terminating client downloading the file from the HTTP Content Server using the received file location URI, optionally via a Localisation Function.

#### 3.2.5.2 Configuration and capability exchange

The client is authorised for the use of the File Transfer if the value configuration parameter FT AUTH defined in section A.1.4 is set to "1".

If the client is authorised for File Transfer, then the client shall advertise the support of File Transfer via

- the IARI media feature tag for File Transfer added in the IMS registration in accordance with the definitions in section 2.4.4,
- the capability discovery of File Transfer as defined in section 2.6.1.3,
- the "+g.3gpp.iari-ref" media feature tag value "urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.fthttp" added in the Contact header of the SIP INVITE requests and the 200 OK responses for sessions related to Group Chat.
- the indication of the content-type *application/vnd.gsma.rcs-ft-http+xml* in the SDP a=accept-wrapped-types attribute of the SIP INVITE requests or the 200 OK response for sessions related to 1-to-1 Chat, Large Message Mode Standalone Messaging and Group Chat.

Otherwise, if File Transfer is not authorised, then the client shall not advertise the support of File Transfer as described above.

The Messaging Server supporting File Transfer shall

- insert in the "+g.3gpp.iari-ref" media feature tag the value "urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.fthttp" in the Contact header of the SIP INVITE request or the 200 OK response for sessions related to Group Chat.
- indicate the content-type *application/vnd.gsma.rcs-ft-http+xml* in the SDP a=accept-wrapped-types attribute of the SIP INVITE requests or the 200 OK response for sessions related to 1-to-1 Chat, Large Message Mode Standalone Messaging and Group Chat.

For the client File Transfer is available in a 1-to-1 Chat conversation, if:

- The *application/vnd.gsma.rcs-ft-http+xml* content type is indicated in the a=accept-wrapped-types attribute from the messaging server during the SDP negotiation and
- the recipient is known to support File Transfer based on capability discovery.

For the client File Transfer is available for 1-to-1 Standalone Messaging, if

- the recipients are known to support the File Transfer capability based on a capability discovery.

For the client File Transfer is available for a Group Chat conversation, if

- the *application/vnd.gsma.rcs-ft-http+xml* content type is indicated in the a=accept-wrapped-types attribute by the messaging server in SDP, and
- the "+g.3gpp.iari-ref" media feature tag value "urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.fthttp" in present the Contact header of the SIP INVITE or SIP 200 OK response received from the Messaging Server for set-up of a Group Chat session.

The Messaging Server shall not forward a message with a File Transfer message body to a recipient client which did not indicate the content-type *application/vnd.gsma.rcs-ft-http+xml* in the SDP a=accept-wrapped-types attribute during session set-up. .

#### 3.2.5.3 File transfer procedure

##### 3.2.5.3.1 Sender procedures

###### 3.2.5.3.1.1 Procedures for Upload and sending

1. To initiate a File Transfer, the sending client shall first send a HTTP POST[14] request without a body to the HTTP Content Server using the value of the FT HTTP CS URI configuration parameter defined in section A.1.4. If the client supports authentication with an GBA bootstrapped security association as defined in [3GPP TS 33.220] it shall indicate this by addition of a GBA product token in the User-Agent header as defined in [3GPP TS 24.109].

[14] 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.

2. The client shall continue processing depending on the received response from the HTTP Content Server:

    a) HTTP 401 AUTHENTICATION REQUIRED error response carrying a WWW-Authenticate header field as defined in [RFC7235] if authentication is required.
    If the client and the service provider's HTTP Content Server supports GBA based authentication then the server returns an HTTP 401 AUTHENTICATION REQUIRED response with an WWW Authenticate header instructing the client to use HTTP digest Authentication with a bootstrapped security association. In this case, the client shall authenticate with the bootstrapped security association as defined in [3GPP TS 24.109]. If the client has no bootstrapped security association in place, the client shall invoke the bootstrapping procedure defined in [3GPP TS 24.109].
    Otherwise, the client shall authenticate using the values of FT HTTP CS USER and FT HTTP CS PWD from the device configuration as defined in Table 86 in section A.1.4.
    b) HTTP 204 NO CONTENT response, the client shall continue processing without authentication.
    c) HTTP 503 INTERNAL ERROR with retry-after header if the server is busy and cannot handle the request, the RCS client shall retry to upload after the time specified in the retry-after header.
    d) A HTTP 302 FOUND response, the client shall change the type of the request to HTTP GET and follow the procedures for OpenID Connect based authentication as defined in section 2.12.2. The procedure results in a reconnection back to the HTTP content server commencing in step 2 of this procedure.
    e) A HTTP 403 FORBIDDEN response, the client is not authorised to apply the File Transfer procedures and shall disable File Transfer in accordance with the definition in section 3.2.5.2.
    f) Any other response, the RCS client shall retry the request.

3. The sender generates a HTTP POST request to upload the file to the HTTP Content Server by making as follows:

    The client shall create the following elements for the transfer of the file to the HTTP Content Server.

    - A File Transfer Transaction ID (TID): this TID value shall be a unique ID generated by the client according to [RFC4122] section 4.2;
    - The thumbnail content: The thumbnail is optional as it is only required for images and videos. The size of this thumbnail shall be smaller than 10 kByte;

        - 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 procedure describing how to create the thumbnail itself, in its raw binary form, is out of scope of this specification.

- The size of a thumbnail should be restricted to the minimum number of octets that provide significance.

- The file content.

The client shall use the elements defined above to create a HTTP POST content body using the *multipart/form-data* content-type to encapsulate the following parts in the listed order:

- An mandatory form element containing the transaction ID:

```
Content-Disposition: form-data; name="tid”
Content-Type: text/plain

<Transaction-ID generated by the client>
```

**Table 19: First form of the HTTP POST method request to upload the file to the HTTP Content Server (Transaction ID)**

- An optional form element containing the thumbnail file content:

```
Content-Disposition: form-data; name="Thumbnail"; filename="<local_filename>"
Content-Type: [mime type depending on the thumbnail; e.g. image/jpeg]

<Thumbnail content>
```

**Table 20: Second form of the HTTP POST method request to upload the file to the HTTP Content Server (Thumbnail contents)**

- A mandatory form element containing the file content:

```
Content-Disposition: form-data; name="File"; filename="<local_filename>"
Content-Type: [mime type depending on the file; e.g. image/jpeg]

<file content>
```

**Table 21: Third form of the HTTP POST method request to upload the file to the HTTP Content Server (file contents)**

The client should include the Content-Length header to indicate the size of the HTTP request body, as described in [RFC7230]. If present, the Content-Length shall indicate the size of HTTP POST body part, i.e. the multipart/form-data entity body.

If the client was requested to authenticate the user in the previous step, then the client shall include an Authorization header as follows:

- If the client was requested to authenticate the user via the values of the configuration parameters FT HTTP CS USER and FT HTTP CS PWD, then the client shall add an Authorization header to the HTTP POST request in accordance with the requested authentication scheme as per [RFC2716] using the *FT HTTP CS USER* and *FT HTTP CS PWD* configuration parameters as credentials. An RCS client shall include the qop directive if provided by the server.

- If the client was requested to authenticate the user via a bootstrapped security association, then the client shall use the stored key material and the B-TID to generate keys specific to the HTTP Content Server as defined in [3GPP TS 33.220]. The client shall add an Authorization header to the HTTP POST request generated from the key material and the B-TID.

The client shall send the HTTP POST request using the HTTP Content Server URL derived from the client configuration parameter FT HTTP CS URI defined in section A.1.4.

4. The following cases apply for the result returned by the HTTP Content Server:

   a) If the upload is successful, the client shall get a HTTP 200 OK response containing a XML in the body that specifies:

      i. The Uniform Resource Locator (URL), size, content type and validity for the thumbnail, if applicable.
      ii. The URLs, 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"
        xmlns:x="urn:gsma:params:xml:ns:rcs:rcs:up:fthttpext">
        <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]"/>
                <x:branded-url>[alternative branded HTTP URL of the file]</x:branded-url>
        </file-info>
</file>
```

**Table 22: HTTP Content Server response: XML contained in the body**

Please note that referring to the XML body in Table 22:

- 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) time zone (e.g. 2007-04-05T14:30:00Z). The validity depends on the configuration the originating Service Provider has set on the HTTP Content Server.

   b) If the upload is not successful, then:

      i. if the HTTP Content Server returns a HTTP 401 AUTHENTICATION REQUIRED and if the client has requested authentication via the

values of client configuration parameters FT HTTP CS USER and FT HTTP CS PWD, then the client shall trigger a configuration request to the configuration server via the procedures defined in section 2.3.2.

ii. if the HTTP Content Server returns a HTTP 401 AUTHENTICATION REQUIRED and if the client has requested authentication via the bootstrapped security association then the client shall invoke the bootstrapping procedure defined in [3GPP TS 24.109].

iii. if the server is busy and cannot handle the request, a HTTP 503 INTERNAL ERROR with retry-after header. The client shall retry to upload after the time specified in the retry-after header for a maximum of three (3) times.

iv. if any other error is returned, then the client shall automatically retry the upload as described in section 3.2.5.3.1.2.

5. If the upload in step 4 was successful, the sender shall process the HTTP Content Server response body and add the additional *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, the receiver will interpret the value as *attachment*. A non-normative HTTP message body content is shown in Table 23.

```
<?xml version="1.0" encoding="UTF-8"?>
<file xmlns="urn:gsma:params:xml:ns:rcs:rcs:fthttp"
        xmlns:x="urn:gsma:params:xml:ns:rcs:rcs:up:fthttpext">
        <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]"/>
                <x:branded-url>[alternative branded HTTP URL of the file]</x:branded-url>
        </file-info>
</file>
```

**Table 23: File Transfer message body content**

The client shall send the File Transfer message body to the receiver(s) in the CPIM message body via the applicable transport service. The client shall use the content-type defined for the File Transfer message body, i.e. *application/vnd.gsma.rcs-ft-http+xml*.

If sending to a single recipient, then there are the following possible scenarios:

- If there is a 1-to-1 chat session established with the user and File Transfer is supported in the session as described in section 3.2.5.2, the session shall be reused to convey the File Transfer message body content in a Chat message.
- If there is no 1-to-1 Chat session established, then the client shall apply the technology selection for 1-to-1 messaging as defined in section 3.2.1. If the client

establishes a session to transmit the File Transfer message body via Standalone Messaging, then the client shall include a dedicated Accept-Contact header field that includes the File Transfer IARI tag defined in section 2.6.1.3 along with *require* and *explicit* parameters.

If sending to multiple recipients, there are the following possible scenarios:

- If the file is to be transferred in an existing Group Chat, a session exists and File Transfer is supported in the session as described in section 3.2.5.2, the session shall be reused to convey the File Transfer content body in a Chat message.
- If the file is to be transferred to an existing Group Chat and no session exists or no Group Chat exists and all recipients are capable to support Chat, then the client shall (re)started first a Group Chat session and transfer the file, if File Transfer is supported in the session.
- Otherwise, if the RCS client is enabled for Standalone Messaging then the File Transfer content body shall be sent using a Standalone Message to the list of recipients carrying a dedicated Accept-Contact header field that includes the File Transfer IARI tag defined in section 2.6.1.3 along with *require* and *explicit* parameters.
- Otherwise, if the client is not enabled for Standalone Messaging, then the client shall send the file via multiple 1-to1 messages.

6. IMDN delivery and display notifications apply as defined for the transport services applicable for File Transfer.

Figure 8 provides a summary overview of the File Transfer sender procedures.

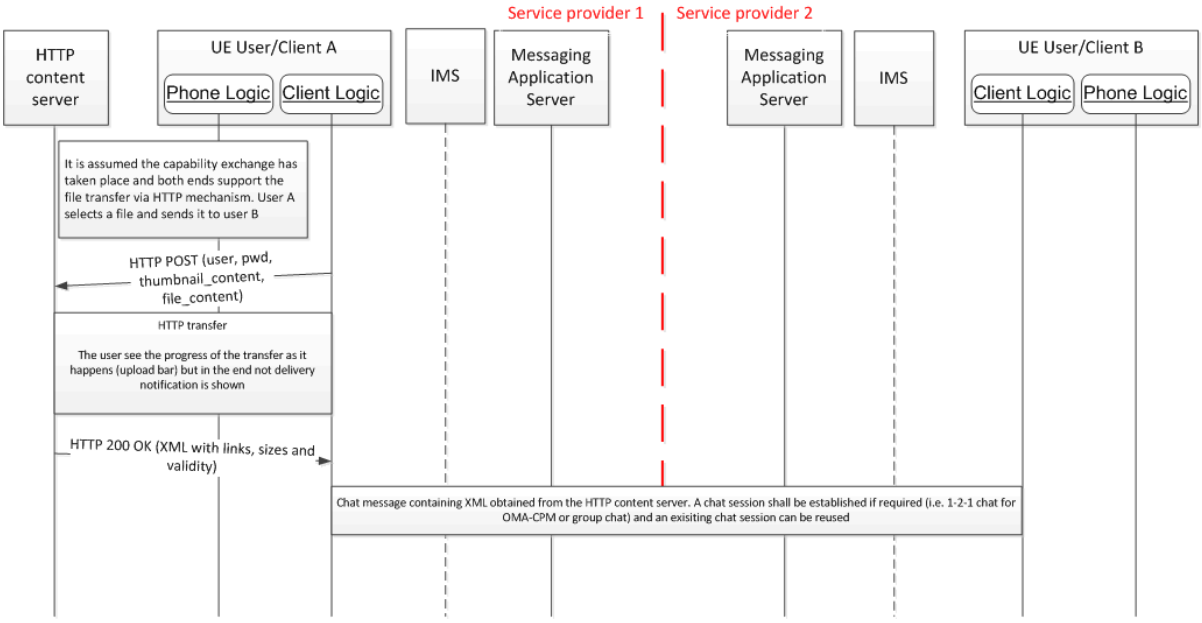



**Figure 8: File Transfer: Sender procedures**

#### 3.2.5.3.1.2 Upload Resume Procedure

In case a file upload cannot be completed, e.g. because the file sender loses network coverage, the client shall 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 HTTP Content Server shall store partial uploads and make them accessible via the related TID defined in 3.2.5.3.1. A Service Provider policy to remove partially uploaded files after some time may apply, thus the resume upload may be possible only for a limited time after the last upload or upload resume attempt.

Overall the client shall retry per file upload up to a maximum of three (3) times after which the upload cannot be resumed and the complete file needs to be uploaded again following the procedure in section 3.2.5.3.1. The following procedure shall be used to resume a failed upload request:

1. A client that intends to resume the upload of an interrupted File Transfer shall first fetch the upload information of the file by sending a HTTP GET request using the HTTP Content Server URI derived from the configuration parameter FT HTTP CS URI defined in section A.1.4 by appending to the query component of the URI the parameters defined below using the application/x-www-form-urlencoded format as defined in [HTML-4.0]. If no query component exists, the client shall add one first, in accordance with the definitions in [RFC3986]. The client shall add

   - a "tid" parameter. The "tid" parameter value shall contain the value of the TID related assigned by the client in the initial upload request (see section 3.2.5.3.1.1).
   - a "get_upload_info" parameter with no value.

   A non-normative example is given in the following:

   - If the value of the configuration parameter FT HTTP CS URI is set to the value `https://upload.operator.com/upload?parm1=foo&parm2=bar`
   - and the value of the TID assigned by the client in the initial upload request is "0815"

   then the client shall use the following URI to construct the HTTP GET request:
   `https://upload.operator.com/upload?parm1=foo&parm2=bar&tid=0815&get_upload_info`

2. 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 24: 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 25: File Transfer upload information schema**

If the client receives a HTTP 200 OK response, 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 3.
- 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 3.2.5.3.1.

NOTE: The file-range refers to the part of the file that has been uploaded prior to the resume upload.

If the client receives a HTTP 404 NOT FOUND or 410 GONE response, then the resume upload cannot be performed (e.g. because the partial files are no longer available). The client shall start the upload procedure beginning using the HTTP POST request as described section 3.2.5.3.1.1.

If the client receives a HTTP 401 AUTHENTICATION REQUIRED with an WWW Authenticate header instructing the client to use HTTP digest authentication with a

bootstrapped required indication in accordance with [3GPP TS 24.109], then the client shall perform the procedures as defined for authentication with a bootstrapped security association.

If the client receives a HTTP 401 AUTHENTICATION REQUIRED with an WWW Authenticate header instructing the client to use HTTP digest authentication without a bootstrapped required indication, or instruction the client to use HTTP Basic authentication, then the client shall perform the procedures as defined for HTTP digest or HTTP basic authentication using the values of the configuration parameters FT HTTP CS USER and FT HTTP CS PWD from the device configuration as defined in in section A.1.4.

If the client receives a HTTP 302 FOUND response, the client shall follow the procedures for OpenID Connect based authentication as defined in section 2.12.2. The procedure results in a reconnection back to the HTTP content server commencing in step 2 of this procedure.

If the client receives a HTTP 403 FORBIDDEN response, then the client is not authorised to apply the File Transfer procedures and shall disable File Transfer in accordance with the definition in section 3.2.5.2.

If the “**Get upload info**" request fails with any other error response, then the client shall retry the “**Get upload info**" request.

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. Then the resume upload cannot be performed (e.g. because the partial files are no longer available). The client shall start the upload procedure beginning using the HTTP POST request as described section 3.2.5.3.1.1.

3. **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 HTTP 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 [RFC7233] is added to the request:

```
PUT <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 26: File Transfer via HTTP upload information content**

The client shall ensure that the file content related to the TID has not been changed between the initial HTTP POST request and the resume upload operation. When the server receives the partial file, it shall append the data according to the Content-Range header. If the upload is successful, a HTTP 200 OK response without body is returned.

If the client receives a HTTP 401 AUTHENTICATION REQUIRED with an WWW Authenticate header instructing the client to use HTTP digest authentication with a bootstrapped required indication, then the client shall perform the procedures as defined for authentication with a bootstrapped security association.

If the client receives a HTTP 401 AUTHENTICATION REQUIRED with an WWW Authenticate header instructing the client to use HTTP digest authentication without a bootstrapped required indication, or instruction the client to use HTTP Basic authentication, then the client shall perform the procedures as defined for HTTP digest or HTTP basic authentication using the values of the configuration parameters FT HTTP CS USER and FT HTTP CS PWD from the device configuration as defined in in section A.1.4

If the client receives a HTTP 302 FOUND response, the client shall change the type of the request to HTTP GET and shall follow the procedures for OpenID Connect based authentication as defined in section 2.12.2. The procedure results in a reconnection back to the HTTP content server commencing in step 3 of this procedure.

If the client receives a HTTP 403 FORBIDDEN response, then the client is not authorised to apply the File Transfer procedures and shall disable File Transfer in accordance with the definition in section 3.2.5.2.

If the "**Resume upload**" request fails with any other response, then the client shall retry by restarting the Upload Resume procedure defined in this section.

The "Resume upload" request can fail due to loss of network coverage. In that case, the operations 1 and 2 shall be repeated with the same TID. In that case, the file-range tag returned from the HTTP content server indicates the sum of all the data uploaded in the uploaded resumes that have taken place so far.

4. **Get download info**: To get the HTTP content body for the complete file to be sent to the file receiver according to the definition in section 3.2.5.3.1.1, the client shall send a HTTP GET request using the HTTP Content Server URI derived from the configuration parameter FT HTTP CS URI defined in section A.1.4 by appending to the query component of the URI the parameters defined below using the application/x-www-form-urlencoded format as defined in [HTML-4.0]. If no query component exists, the client shall add one first, in accordance with the definitions in [RFC3986]. The client shall add:

   - a "tid" parameter. The "tid" parameter value shall contain the value of the TID assigned by the client in the initial upload request (see section 3.2.5.3.1.1).

- a "get_download_info" parameter with no value

A non-normative example is shown below:

```
https://upload.operator.com/upload?
      parm1=foo&parm2=bar&tid=0815&get_download_info
```

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 22).

If the server sends back a HTTP 401 AUTHENTICATION REQUIRED with an WWW Authenticate header instructing the client to use HTTP digest authentication with a bootstrapped required indication, then the client shall perform the procedures as defined for authentication with a bootstrapped security association.

If the server sends back a HTTP 401 AUTHENTICATION REQUIRED with an WWW Authenticate header instructing the client to use HTTP digest authentication without a bootstrapped required indication, or instruction the client to use HTTP Basic authentication, then the client shall perform the procedures as defined for HTTP digest or HTTP basic authentication using the values of the configuration parameters FT HTTP CS USER and FT HTTP CS PWD from the device configuration as defined in in section A.1.4.

If the server sends back a HTTP 302 FOUND response, the client shall follow the procedures for OpenID Connect based authentication as defined in section 2.12.2. The procedure results in a reconnection back to the HTTP content server commencing in step 4 of this procedure.

If the server sends back a HTTP 403 FORBIDDEN response, then the client is not authorised to apply the File Transfer procedures and shall disable File Transfer in accordance with the definition in section 3.2.5.2.

If the server sends back any other HTTP error response, then the "**Get download info**" request failed. The client shall retry by restarting the Upload Resume procedure defined in this section.

The whole procedure (including the initial upload is summarized in following figures:

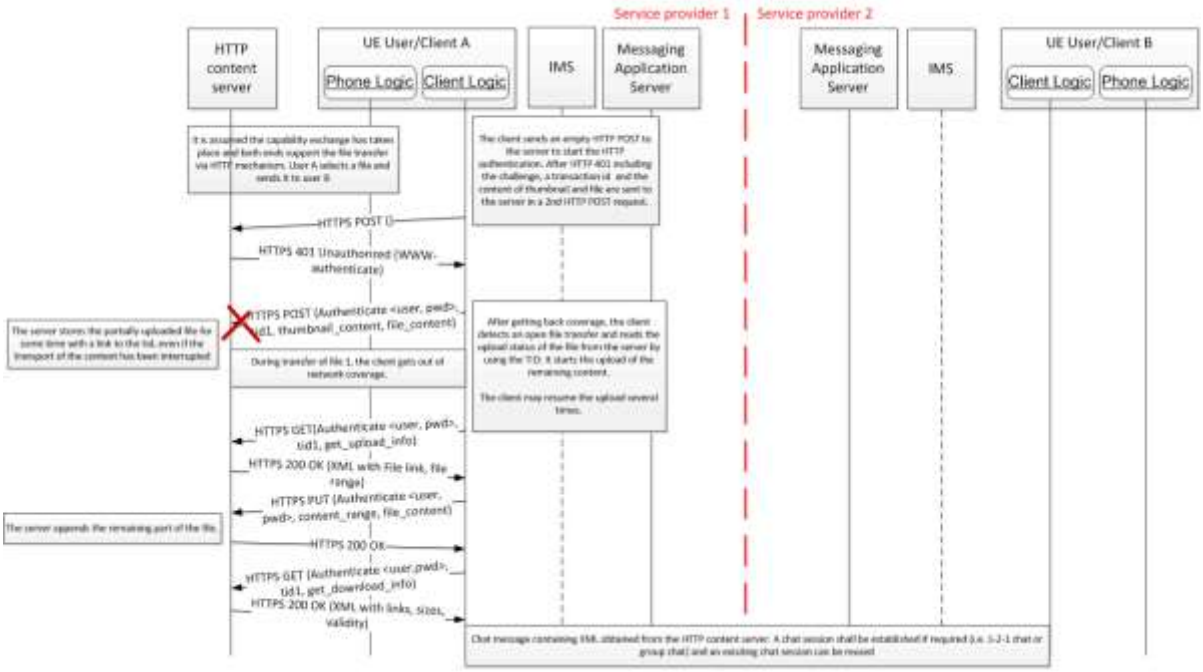



**Figure 9: File Transfer: Resume upload**

In case the resume is not possible (anymore), the flow shall be as follows:

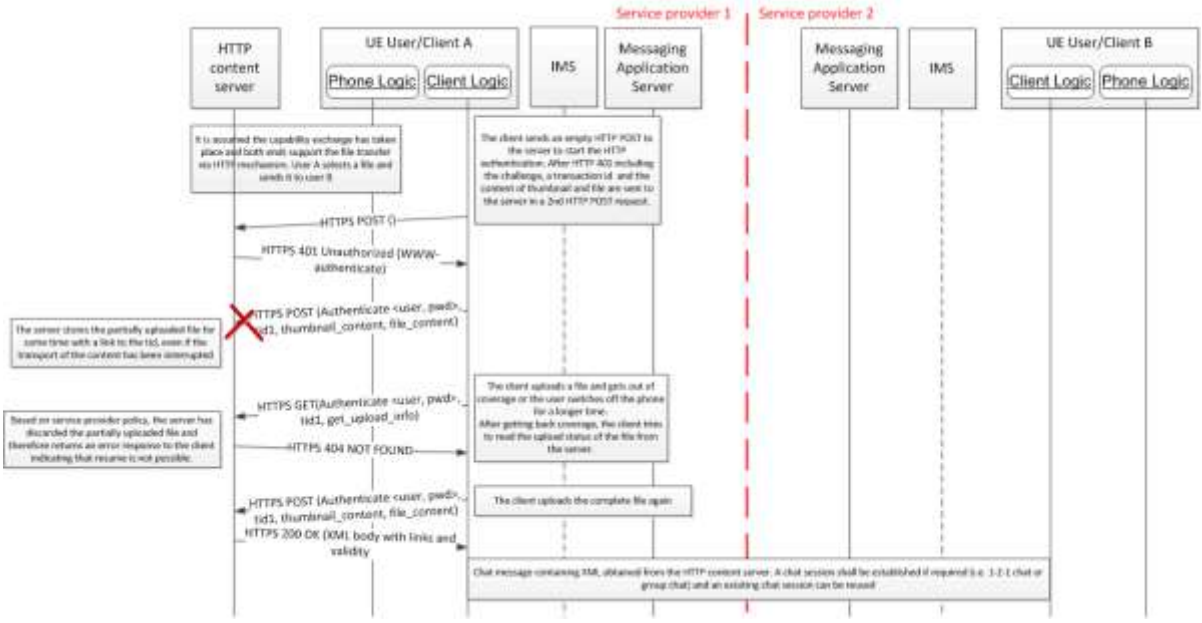



**Figure 10: File Transfer: Resume upload not possible**

#### 3.2.5.3.2 Receiver procedures

##### 3.2.5.3.2.1 File download procedure

If the client receives a Chat or Standalone Message with a File Transfer message body, then the client shall apply delivery disposition notifications as defined for the corresponding transport service.

Then, the RCS shall:

1. if present, download the thumbnail via the download procedure defined below.
2. if the user accepts the download or File Transfer auto-accept applies, shall download the file via the download procedure below.

NOTE: The procedures defined in this section apply also for retrieval of files in result of procedures related to multi device and the Common Message Store, see section 4.1.15.3.2.

To download a given file, the client shall

1. if value of the configuration parameter FT HTTP DL URI defined in section A.1.4 is present, create the download URI by appending to the query component of the URL contained in the configuration parameter the request parameters defined below using the application/x-www-form-urlencoded format as defined in [HTML-4.0]. If no query component exists, the client shall add one first, in accordance with the definitions in [RFC3986]. The client shall add:

   - a mandatory "url" parameter with the file URI taken from the File Transfer message body or any other content URL the client received via RCS messaging.
   - an optional "id" parameter with the message-ID taken from the IMDN message-ID of the Chat or Standalone Message or taken from the message object of the Common Message Store. The "id" parameter shall be present if the content URL was received via Chat or Standalone messaging or via a message object stored in the Common Message Store.
   - an optional "op" parameter. The parameter shall be present if the File Transfer message body or any other content URL was received in a 1-to-1 RCS messaging conversations. The parameter value shall contain the address of the other party of the message in the 1-to-1 conversation. The client shall derive the value from the SIP signalling or the address header of the Common Message Store for a 1-to-1 Chat Message or a Standalone Message. If the message was received from another user as indicated by the value or absence via of the CPIM "Message-Direction" header defined in section C.1.9 of [RCS-CPM-CONVFUNC-ENDORS] or the "Direction" attribute defined in [CPM-MSGSTOR-REST], then it shall be taken from the authenticated originator address of the message. If the message was sent by the own user as indicated via the CPIM "Message-Direction" header defined in section C.1.9 of [RCS-CPM-CONVFUNC-ENDORS] or the "Direction" attribute defined in [CPM-MSGSTOR-REST], then it shall be taken from the authenticated recipient address of the message. The value of the authenticated originator or recipient address shall be used unaltered.
   - an optional "ci" parameter. The parameter shall be present if the File Transfer message body was received in a Group Chat. The parameter value shall contain the value of the conversation-id taken from SIP signalling of a Group Chat session or from message object of a Group Chat message in the common message store.

   Example: If

   - the value of the configuration parameter FT HTTP DL URI is set to *https://dl.operator.com/path?parm=foo*

- and if the download URI in the File Transfer message body of a 1-to-1 Chat message is:
  *https://ftcontentserver.rcs.mnc001.mcc262.pub.3gppnetwork.org/dl?uid=1234*
- and the message ID of the chat message is *123456789*
- and if the authenticated originator address of the of a 1-to-1 Chat message is *sip:+491711234567@ims.mnc001.mcc262.3gppnetwork.org;user=phone*

then the client's download URI results in
*https://dl.operator.com/path?parm=foo&url=https%3A%2F%2Fftcontentserver.rcs.mnc001.mcc262.pub.3gppnetwork.org%2Fdl%3Fuid%3D1234&id=123456789&op=sip%3A%2B491711234567%40ims.mnc001.mcc262.3gppnetwork.org%3Buser%3Dphone*

2. otherwise, use the file URI received in the File Transfer message body
3. if the client supports authentication with an GBA bootstrapped security association as defined in [3GPP TS 33.220], shall indicate this by addition of a GBA product token in the User-Agent header as defined in [3GPP TS 24.109]
4. send a HTTP GET request using the derived file URI.
5. If the client receives in result of the processing of the request from the HTTP Content Server

   - a HTTP 401 AUTHENTICATION REQUIRED response with an WWW Authenticate header instructing the client to use HTTP digest Authentication with a bootstrapped security association as defined in [3GPP TS 24.109].

     If the client has no bootstrapped security association in place it shall invoke the bootstrapping procedure defined in [3GPP TS 24.109].

     The client shall use the stored key material and the B-TID to generate keys specific to the HTTP Content Server as defined in [3GPP TS 33.220]. The client shall add an Authorization header generated from the key material and the B-TID and send the HTTP GET request for authentication.

     If a HTTP 401 AUTHENTICATION REQUIRED response is received for the HTTP GET request, then the client shall invoke the bootstrapping procedure defined in [3GPP TS 24.109].

   - a HTTP 401 AUTHENTICATION REQUIRED response with an WWW Authenticate header without a "bootstrapping required" indication, then the client shall create an authorization header from the values of FT HTTP CS USER and FT HTTP CS PWD from the device configuration as defined in in section A.1.4 and send the HTTP GET request for authentication.

     If a HTTP 401 AUTHENTICATION REQUIRED is received in the result of the HTTP GET request then the client shall trigger a configuration request to the configuration server via the procedures defined in section 2.3.2.

   - a HTTP 302 FOUND response, the client shall follow the procedures for OpenID Connect based authentication as defined in section 2.12.2. The procedure results

in a reconnection back to the HTTP content server commencing in the processing of the HTTP content server response.

- If the client receives a HTTP 403 FORBIDDEN response, then the client is not authorised to apply the File Transfer procedures and shall disable File Transfer in accordance with the definition in section 3.2.5.2.
- a HTTP 503 INTERNAL SERVER ERROR with a Retry-After header, then the client shall retry, the recommended value to retry will be specified in the “Retry-After” header.
- a HTTP 404 NOT FOUND or HTTP 410 GONE, then the client shall stop the file download procedure and inform the user that the file is no longer available.
- any other error, then 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 [RFC7233] may be used to obtain the remaining part of the file.
- a HTTP 200 OK response with a file object in the body then client shall handle the file according to the content-type received in the HTTP GET response or in the File Transfer message body and the file-disposition attribute received in the File Transfer message body.

6. The client shall follow the service definitions of the File Transfer service for the handling of display disposition notifications.

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.

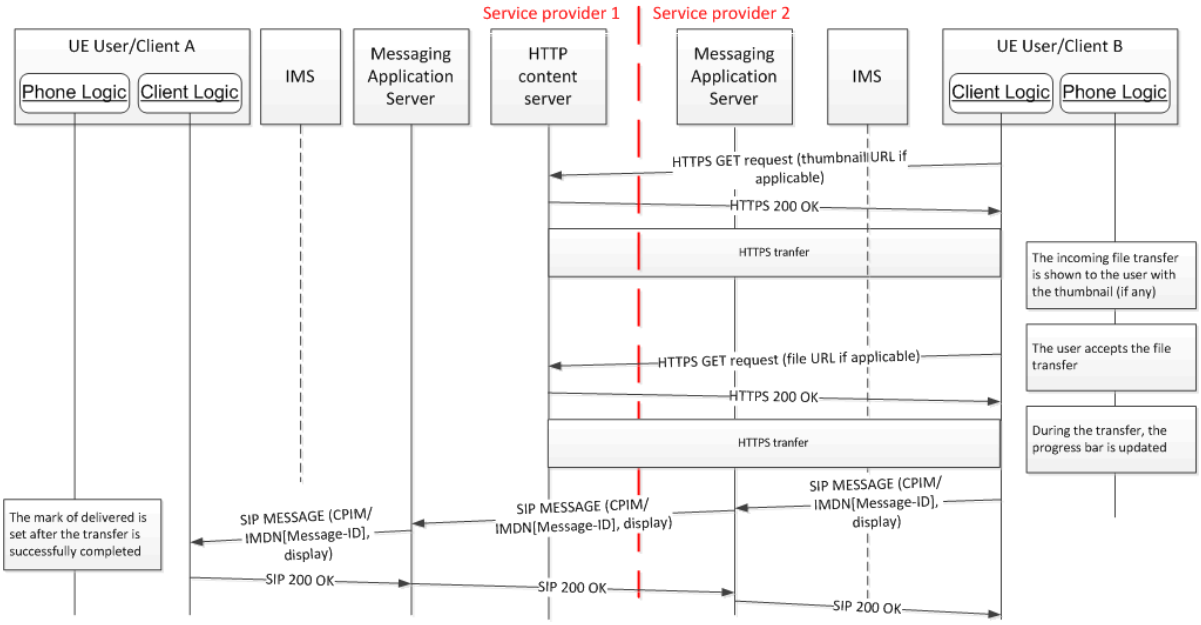



**Figure 11: File Transfer via HTTP: Receiver procedures**

#### 3.2.5.3.2.2 File Transfer auto-accept

Consistently with Annex A sections A.1.4 and A.2.4, if the parameter *FT AUT ACCEPT* is set to 1 and the file size indicated in the File Transfer message body 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.

#### 3.2.5.4 Schema Definition

Both the HTTP Content Server response and the File Transfer message body transferred between the clients shall comply with the following XML Schema. The schema is extensible via the standard schema extension mechanism. Clients receiving unsupported elements or attributes shall ignore them.

The content-type assigned to the schema of the File Transfer body shall be *application/vnd.gsma.rcs-ft-http+xml*.

```
<?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 27: File transfer message body schema**

The schema defined in Table 27 is extended to enable the HTTP Content Server to assign a user friendly URL to a file. The user friendly URL shall be conveyed by the HTTP Content Server to the client via the "branded-url" element defined in the schema in Table 28.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:gsma:params:xml:ns:rcs:rcs:up:fthttpext"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns="urn:gsma:params:xml:ns:rcs:rcs:up:fthttpext"
        elementFormDefault="qualified"
        attributeFormDefault="unqualified">
        <xs:element name="branded-url" type="xs:anyURI"/>
</xs:schema>
```

**Table 28: File Transfer Message Body schema Extension**

The "branded-url" element shall be added, if available on the HTTP Content Server, as an extension to the "data" element of the "file info" element of the HTTP Message Body schema defined in Table 27.

Clients receiving the "branded-url" element in an HTTP Content Server response body shall forward the element unaltered to recipients when using the HTTP message body.

The nature and structure of the URL value in the "branded-url" element is left to the discretion of the service provider of the HTTP Content Server.

#### 3.2.5.5 HTTP Content Server URL

For the composition of the URL to locate files on the HTTP Content Server, the requirements of this section apply.

To enable the traceability of the HTTP transactions between operators over the NNI, the URL to locate files on the HTTP Content Server shall include the FQDN as defined in Table 29.

| ftcontentserver.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org |
|---|

**Table 29: HTTP Content Server FQDN**

Apart from the provider specific path and query elements of the HTTP Content Server URL to locate files on the HTTP Content Server, the URL may contain the following well-known URL parameters to provide meta-information describing the file. The URL parameters are appended to the query component of the HTTP Content Server URL using the application/x-www-form-urlencoded format as defined in [HTML-4.0].

Table 30 defines the parameters to provide additional meta-data describing the file for File Transfer fallback.

| Parameter | Type | Value |
|---|---|---|
| s | Integer | Identifies the size of the file in bytes<br><br>The presence of the parameter is mandatory, if the URL locating a file on a HTTP Content Server contains additional meta data describing the file. |
| t | String | Value of the Multipurpose Internet Mail Extensions (MIME) content-type header of the file as defined in [RFC2045].<br>Note: reserved characters in the content-type header value have to be represented using percent encoding in accordance with [RFC3986].<br><br>The presence of the parameter is mandatory, if the URL locating a file on a HTTP Content Server contains additional meta data describing the file. |
| e | String | Combined date and time in UTC time zone in ISO8601 basic format, i.e. YYYYMMDDThhmmssZ.<br>It indicates the date and time of expiry of the file, e.g. 20170419T135227Z<br><br>The presence of the parameter is mandatory, if the URL locating a file on a HTTP Content Server contains additional meta data describing the file. |
| d | Integer | Playing length in seconds of the Audio Message. It indicates that the file located via the URL is a RCS Recorded Audio Message as defined in section 3.2.7 that can be played directly from the Chat application upon user action.<br><br>The parameter is optional in the URL location a file on a HTTP Content Server. The parameter shall not be present in the HTTP Content Server response body for a URL locating a file. |

**Table 30: HTTP URL parameters for File Transfer fallback**

Service providers should add the parameters in the HTTP Content Server response body defined in section 3.2.5.3.1 to the value of the "url" attribute of the "data" element of the "file-info" element with the "type" attribute set to "file". It is recommended that implementations ensure that the maximum length of the URL locating a file on a HTTP Content Server does not exceed the length of the user data of one SMS message.

#### 3.2.5.6 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 recognised CA. That is, the root certificates are similar to those used by standard webservers which are widely recognised by browsers and web-runtime implementations in both PCs and devices.

#### 3.2.5.7 File Transfer Fallback

##### 3.2.5.7.1 Introduction

The procedures for File Transfer fallback defined in this section enable RCS clients to provide a File Transfer experience if RCS is not available end-to-end.

The procedures for File Transfer fallback are applicable if

- a File Transfer is subject to delivery assurance via client fallback to SMS or
- a file is transmitted to a non-RCS capable recipient or a RCS capable recipient without File Transfer capability.

The procedures for File Transfer fallback rely on transport of File Transfer specific content via SMS messages.

##### 3.2.5.7.2 File Transfer via SMS Capability

Section 2.6.1.3 defines the client capability for the support of the File Transfer via SMS service. A client supporting File Transfer and either SMS or Standalone Messaging or both shall advertise the capability

- via the media feature tag defined in Table 9 for File Transfer via SMS in SIP OPTIONS requests and responses for capability discovery or
- via the service description defined in Table 9 for File Transfer via SMS in presence documents for capability discovery

in accordance with the definitions for capability discovery defined in section 2.6.

##### 3.2.5.7.3 Sender Procedures

The procedures defined in this section apply

- if the user requests to send a file to a recipient with no RCS capabilities or a recipient with RCS capabilities but without support of the File Transfer via SMS service as indicated by the capability defined in section 3.2.5.7.2. In this case, the user may prefer to send the message via MMS or as a "text message with a link". The operator is able to provide a default selection via the configuration parameter FT HTTP FALLBACK defined in section A.1.4 and A.2.4.
- if a chat message conveying a File Transfer via HTTP is subject to client fallback as defined in section 3.2.3.8.

#### 3.2.5.7.3.1 Sending of a File to a Recipient with no File Transfer Capability

If the user requests to send a file to a recipient with no RCS capabilities or a recipient with RCS capabilities but without support of the File Transfer via SMS service indicated by the capability defined in section 3.2.5.7.2, then

- if the user or the client has selected MMS to transfer the file and the file complies or can be converted to comply with the formats and codecs defined in [MMSCONF], then the client shall compose a MMS message and send it following the procedures defined in [3GPP TS 23.140], otherwise
- the client shall upload the file to the HTTP Content Server as defined in section 3.2.5.3.1.
- If successful, the client shall check whether the HTTP message body received from the HTTP Content Server contained an "branded-url" element in the file-info element with type "file", then
    - if the "branded-url" element is present it shall use the URL contained in its value, otherwise
    - it shall use the URL contained in the value of the "url" attribute of the "data" element of the "file-info" element with type "file",
- compose a new message containing the URL next to some explanatory text indicating the purpose of the message and send it to the recipient via
    - RCS Standalone Messaging, if enabled as defined in section 3.2.2, otherwise
    - via SMS.

#### 3.2.5.7.3.2 File Transfer Client Fallback

If the originating client has uploaded a file to the HTTP Content Server for a File Transfer via HTTP and the client applies monitoring of the delivery of chat messages within the 1-to-1 Chat session in accordance with the definitions in section 3.2.3.8, then the client shall keep the data of the HTTP Content Server response body at least until the delivery of the chat message is confirmed.

Precondition for the application of File Transfer via SMS is that the client has uploaded a file to the HTTP Content Server as defined in section 3.2.5.3.1 and has received and kept the data of the HTTP Content Server response body.

If the originating client decides to fall back to SMS for a File Transfer as result of the procedures for client fallback as defined in section 3.2.3.8.3 and the recipient supports File

Transfer via SMS, as indicated by the capability defined in section 3.2.5.7.2, then the originating client

- shall inspect the URL contained in the "url" attribute of the "data" element of the "file-info" element with type "file" received in the HTTP message body from the HTTP Content Server to determine whether the "s", "t" and "e" URL parameters as defined in section 3.2.5.5 are present, then
- if there is none of these URL parameters present, then the client shall generate

    - a “s” parameter as defined in section 3.2.5.5 using the value extracted from the file-size element of the file-info element with type “file” included in the HTTP message body of the response,
    - a “t” parameter as defined in section 3.2.5.5 using the value extracted from the content-type element of the file-info element with type “file” included in the HTTP message body of the response,
    - an “e” parameter as defined in section 3.2.5.5 using the value extracted from the “until” attribute contained in the data element of the file-info element with type “file” included in the HTTP message body of the response,
    - append it to the query part of the URL taken from "url" attribute of the "data" element of the "file-info" element with type "file" using the application/x-www-form-urlencoded format as defined in [HTML-4.0] respecting the definitions of [RFC3986]. If no query part exists, the client shall create one first,

- otherwise use the URL contained in the "url" attribute of the "data" element of the "file-info" element with type "file" unaltered
- if the file is a RCS Recorded Audio Message (RRAM) as defined in section 3.2.7, then the client shall generate a “d” parameter as defined in section 3.2.5.5 using the “playing-length” of the RRAM as defined in section 3.2.7.2.2 and append it to the URL resulting from previous processing using the application/x-www-form-urlencoded format as defined in [HTML-4.0] respecting the definitions of [RFC3986],
- add the resulting URL to the user data of a SMS message and
- send the SMS message to the recipient address.

If the originating client decides to fall back to SMS for a File Transfer as result of the procedures for client fallback as defined in section 3.2.3.8.2 and the recipient does not support File Transfer via SMS via the capability defined in section 3.2.5.7.2, then the client

- shall check whether the HTTP message body received from the HTTP Content Server contained an "branded-url" element in the file-info element with type "file", then

    - if the "branded-url" element is present it shall use its value, otherwise
    - it shall use the URL contained in the "url" attribute of the "data" element of the "file-info" element with type "file",

- add the URL next to some explanatory text indicating the purpose of the message to the user data of a SMS message and
- send the SMS message to the recipient address.

#### 3.2.5.7.4 Receiver Procedures

On reception of a SMS message, the client shall parse the user data of the message.

If the user data contains a HTTP(s) URL and the FQDN of the URL conforms to the definitions of the HTTP Content Server URI as defined in section 3.2.5.5, then the client shall apply the UX procedures defined for suppression and replacement of the HTTP Content Server URL. The client shall take the URL parameters for File Transfer fallback as defined in section 3.2.5.5 into account.

To retrieve the file using the URL received in the SMS message and if the FQDN conforms to the definitions of the HTTP Content Server URL as defined in section 3.2.5.5, then the client shall apply the file download procedure defined in section 3.2.5.3.2.1. If the configuration parameter FT HTTP DL URI defined in section A.1.4 is present, then the client shall follow the procedure as defined for this case in section 3.2.5.3.2.1 taking the following requirements into account. When creating the download URL, the client

- shall not add the "id" parameter to the download URL
- shall add the "op" parameter containing the address value of the originator address of the received SMS message prefixed by the string "sms:" If the address value contains an international E.164 number, then the address value shall be prefixed with "+"
  Example: if the originator address of the SMS is an international E.164 number with an address value "491711234567", then the resulting "op" parameter is encoded as follows:
  op=sms%3A%2B491711234567
- shall not add the "ci" parameter.

#### 3.2.5.7.5 Network Interworking

The procedures in the network for File Transfer fallback are network internal and therefore outside of the scope of this document. As defined in section 3.2.5.7.2, the client shall add the media feature tag defined in Table 3 for File Transfer via SMS in the Contact header field of the SIP REGISTER allowing the network to detect the client capability. The formats of SMS messages resulting from the network interworking procedures shall follow the formats defined for the client-based fallback defined in section 3.2.5.7.3.

### 3.2.5.8 HTTP State Management

The client shall support for the HTTP procedures for File Transfer the HTTP state management defined in [RFC6265]. This includes all HTTP requests and responses between the client and HTTP Content Servers as part of the File Transfer procedures for upload, download and the corresponding resume procedures. This allows a HTTP Content Server to return in HTTP responses a Set-Cookie header. The client shall apply the parsing and storage procedures of the Set-Cookie header as defined [RFC6265]. It shall send the cookie header in HTTP requests to HTTP Content Servers respecting the cookie attributes provided by the HTTP Content Server in the Set-Cookie header in accordance with [RFC6265].

With this, the HTTP Content Server is able to make use of all the functions of HTTP state management.

#### 3.2.5.9 Handling of specific content

##### 3.2.5.9.1 Personal Card format

There are multiple formats for the transfer of Personal Cards. This section defines the transfer formats and procedures for Personal Cards in RCS.

An RCS compliant device shall support receiving following formats:

- the vCard 2.1 format as defined in [vCard21] and
- the vCard 3.0 format as defined in [RFC2425] and [RFC2426] and
- the vCard 4.0 format as defined in [RFC6350].

NOTE: vCard 3.0 [RFC2425] and [RFC2426] were obsoleted by [RFC6350]. These obsoleted references are used for vCard 3.0 formats that do not comply with the latest RFC.

In addition, an RCS may support receiving the following format

- the Personal Contact Card (PCC) format defined in [CAB_TS].

The vCard 3.0/4.0 format as defined above shall be used for sending.

Variations in the implementation of Personal Card formats may lead to data loss when Personal Cards are exchanged. To limit the effect, the following fields are considered key fields for RCS. No data of these fields should be lost when contact information is exchanged in RCS.

- **Name**: Composed names (such as “Jean-Baptiste”) shall be supported properly.
- **Personal Information**:
  - Nickname
  - Photo
  - Birthdate
  - Comment
- **Telephone numbers**: 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
  - Time zone

Sending and receiving a Contact Card via File Transfer via HTTP or File Transfer fallback is technically the same as sending any other file.

If the format for transferring a Contact Card file is vCard 2.1, vCard 3.0 or vCard 4.0, then the MIME content type "*text/vcard*" shall be used for File Transfer.

If the format for transferring a Contact Card is the CAB (Converged Address Book) 1.0 PCC XML format, then the CAB PCC MIME content type "*application/vnd.oma.cab-pcc+xml*" shall be used for File Transfer.

On the receiving side, if the receiving RCS adds the Contact Card file delivered through File Transfer to the local address book, the receiving RCS client shall apply the mapping of the RCS supported fields between the received format and the used format of the local address book database. For conversion between PCC and vCard formats, refer to section 5.4.3 of [CAB_TS].

If the receiving client does not support the format of a Contact Card, then the client handling for unsupported content types applies.

#### 3.2.5.9.2 Audio Message

The handling of audio messages is described in section 3.2.7.

### 3.2.6 Geolocation Push services

#### 3.2.6.1 Overview

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. The format that shall be used is described in section 3.2.6.5.

In older versions of RCS the CPM File Transfer service (see [RCS-CPM-CONVFUNC-ENDORS]) was 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). To provide backward compatibility to those, an RCS client shall support receiving such location share requests.

#### 3.2.6.2 Geolocation Push Procedure

The Geolocation PUSH service shall send the geolocation information 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 sending of the geolocation information 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 2.6.1.3.

When these conditions are fulfilled, a client can transfer the geolocation information in a CPIM wrapper that is transferred using an MSRP SEND request with the encapsulated Content-type header of the CPIM message set to *application/vnd.gsma.rcspushlocation+xml*.

##### 3.2.6.2.1 1-to-1 Exchange of Geolocation PUSH

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 as specified in section 3.2.3.1. The Geolocation PUSH XML message body itself (i.e. geolocation information in a CPIM wrapper with the encapsulated Content-type header set to *application/vnd.gsma.rcspushlocation+xml*) shall then be sent as first message in the Chat.

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.

##### 3.2.6.2.2 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 2.6.1.3 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 2.6.1.3 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.

#### 3.2.6.3 Geolocation Push Fallback

A client supporting Geolocation PUSH and either SMS or Standalone Messaging shall support Geolocation PUSH fallback to SMS, and thus support the rendering of the user data of a short message for Geolocation Push as defined in section 3.2.6.3.2 and the procedures for "geo" URI defined in section 3.2.6.5.4.

##### 3.2.6.3.1 Sender procedures

If the originating client decides to fallback to SMS for a Geolocation Push message and the recipient supports Geolocation Push via SMS as indicated by the capability defined in section 2.6.1.3, then the sender client shall use the position and the label of the RCS Location information data sent in the Geolocation Push message and generate a "geo" URI as defined in section 3.2.6.5.4.

##### 3.2.6.3.2 Receiver procedures

On reception of a SMS message, the client shall parse the user data of the message.

If the user data contains a "geo" URI as defined in [RFC5870], then the client shall apply the UX procedures defined for suppression and replacement of the "geo" URI string. The client shall apply the rules for the presence and absence of the "label" via the "geo" URI extension defined in section 3.2.6.5.4 in accordance with the definitions of section 3.2.6.5.2.

##### 3.2.6.3.3 Network Procedures for Fall Back for Geolocation Push

The procedures in the network for fallback for Geolocation Push are network internal and therefore outside of the scope of this document. However, to facilitate a solution at a later time, as specified in section 2.4.4.1 the User Equipment (UE) shall include the IMS Application Reference Identifier (IARI) defined in section 2.6.1.3 in the Contact header field of the SIP REGISTER request along with the rest of the feature tags the UE is required to include. The format of the SMS messages sent shall follow the format defined for the client-based fallback described in section 3.2.6.5.4.

##### 3.2.6.3.4 Geolocation Push service Technology Selection

If the RCS Geolocation Push service is enabled (i.e. PROVIDE GEOLOC PUSH is set to 1) and the Chat service is enabled (i.e. CHAT AUTH is set to 1), the client shall follow the procedures defined in this section.

If a message for RCS Geolocation Push is subject to delivery assurance as defined in this document, the client shall apply the procedures defined in this section. If a message for RCS Geolocation Push is subject to delivery assurance as defined in this document and the receiver does not support the

procedures defined in this section, the client shall send the location based on the rules defined in section 3.2.1:

- Selecting an RCS Standalone message if RCS Standalone Messaging is enabled, otherwise
- Selecting an SMS.

If the RCS Geolocation Push service is enabled (i.e. PROVIDE GEOLOC PUSH is set to 1), the RCS 1-to-1 Chat service is disabled (i.e. CHAT AUTH is set to "0") and the RCS Standalone messaging service is enabled, the regular Geolocation Push service defined in earlier in this section cannot be used due to its dependency on Chat. Therefore, the location information shall be sent as a text message based on the technology selection rules described in section 3.2.1. For the format of the message, the procedures defined in this section shall be applied (i.e. it shall depend on whether the recipient indicated the Geolocation Push via SMS capability defined in section 2.6.1.3).

For sending location information towards non-RCS users, the 1-to-1 Messaging technology selection rules towards non-RCS users defined in section 3.2.1 shall apply.

#### 3.2.6.4 Backward compatibility for in-call sharing

To provide backward compatibility to older versions of RCS, a client shall support receiving a location shared during a call with CPM File Transfer. The CPM File Transfer request is routed to the RCS geolocation application (internal routing based on the IARI).

NOTE: This is the only scenario where CPM File Transfer is used in this version of RCS.

On the receiving side, the File Transfer invitation will be automatically accepted.

If the transfer is successful, the application triggers the user in a pop up menu to handle the location information.

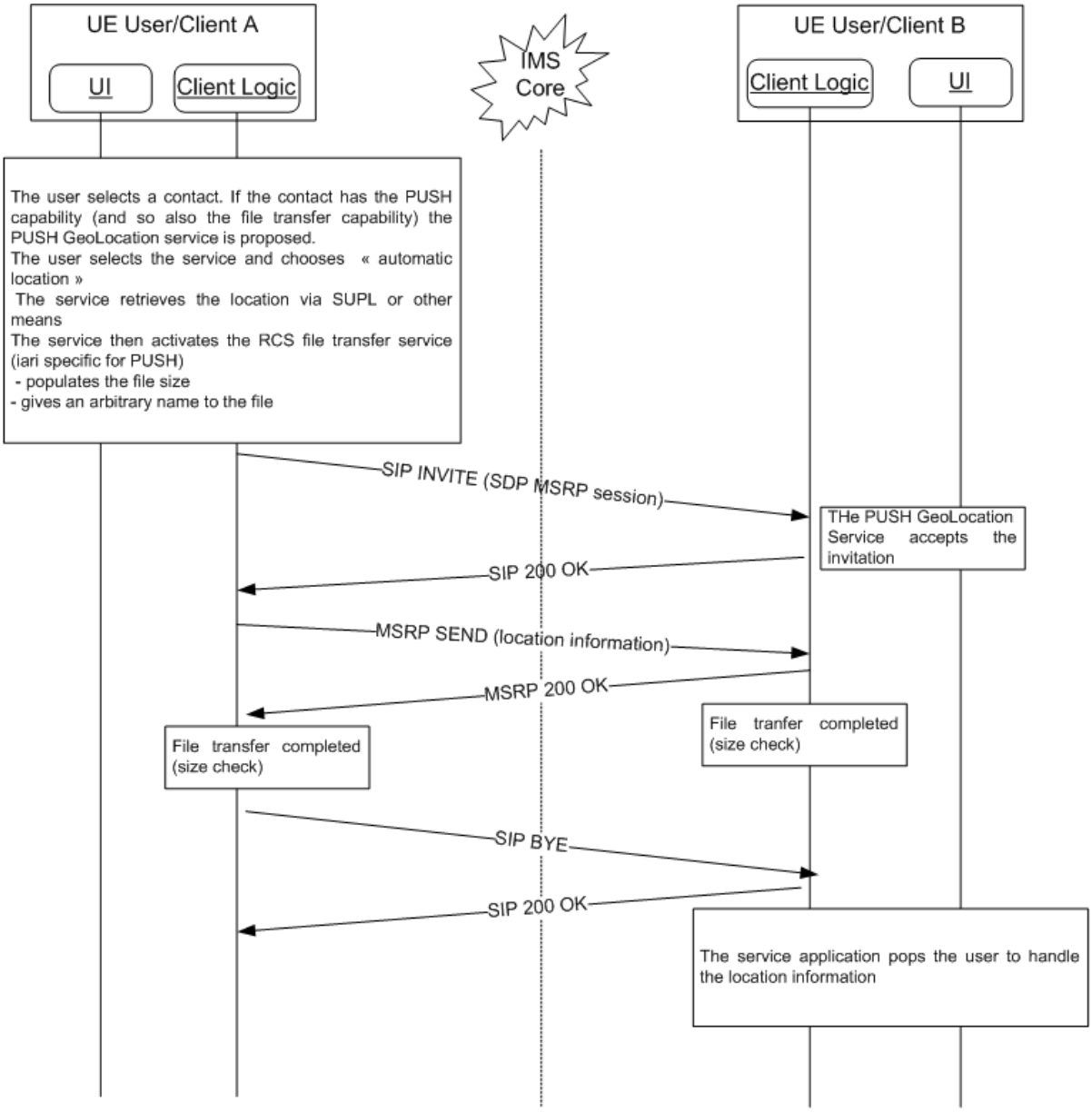



**Figure 12: Backward compatibility: Push of geolocation information during a voice or video call using CPM File Transfer**

### 3.2.6.5 Location Information format

#### 3.2.6.5.1 General

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 31: Geolocation PUSH Envelope XML schema**

#### 3.2.6.5.2 RCSPushLocation data model

| Attribute | Specification | Comment |
|---|---|---|
| Person:<br><rcsenvelope> -><br><rcspushlocation> | Table 31 | Each client only publishes one <rcsenvelope> and one <rcspushlocation> element.<br>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<br><rcspushlocation> -><br><time-offset> | Table 31, [RFC4480] and [Presence2.0_DDS] | The geolocation application may use this element to provide information on the current time zone<br>See following chapter section for more information on the handling of the expiry of this information |
| Geographical Information<br><rcspushlocation> -><br><geopriv> -><br><location-info> -><br><usage-rules> | Table 31, [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.<br>See following section for more details on its encoding and on the handling of the expiry of this information |

| Attribute | Specification | Comment |
|---|---|---|
| Timestamp:<br><rcspushlocation> -><br><timestamp> | Table 31, [RFC4479] | Timestamp when the location information was pushed |

**Table 32: RCSPushLocation data model attributes**

#### 3.2.6.5.3 RCS Location 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 authorised by the Service Provider) an "*until*" attribute in that element with a value provided by the user.

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.

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 European Petrol Survey Group 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 *label* attribute) shall not exceed 200 characters. The text is entered by the user.

An RCS client shall ignore any other type of data provided in the <*location-info*> element.

The EPSG format requires that the coordinate representation be 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 33: Example of location information data**

#### 3.2.6.5.4 Geolocation Push URI for fallback

A client supporting Geolocation Push fallback shall be able to

- generate, using RCS Location information data, and
- resolve and render

a “geo” URI according to [RFC5870].

For the purpose of Geolocation Push fallback, the "geo" URI format of [RFC5870] is extended by a new parameter to carry a "label". The usage of the "label" parameter shall follow the definitions for the "label" in Geolocation Push defined in section 3.2.6.5.2.

"Geo" URI parameters extending [RFC5870] are defined in Table 34.

| Parameter | Value Restriction | Value |
|---|---|---|
| rcs-l | Constrained | Contains an UTF-8 character encoded label text that can be used to tag the nature of the location (e.g. indicate that it is the home or provide an address, name of restaurant, etc.) in the context of Geolocation Push. If the label parameter is absent, the location that is shared is assumed to be the sharing users own position.<br>NOTE: non-ASCII and reserved characters have to be represented using percent encoding in accordance with [RFC5870]. |

**Table 34: "geo" URI Parameter Extensions**

NOTE: It is recommended that implementations ensure that the maximum length of the URLs does not exceed the length of the user data of one short message.

Example "geo" URI with parameter extension:

```
geo:50.7311865,7.0914591;u=10;rcs-l=The%20Quiet%20Man%20%F0%9F%8D%BB
```

### 3.2.7 Audio Messaging

#### 3.2.7.1 Overview

An RCS client shall encode the audio message using the Adaptive Multi-Rate (AMR) codec.

The RCS Recorded Audio Message (RRAM) shall be formatted in the file format defined in [RFC4867].

The transport of RRAM uses the File Transfer as defined in section 3.2.5. The following features are applicable for Audio Messaging:

- disposition notifications of the File Transfer transport services
- store and forward of the File Transfer transport services
- auto-acceptance rules for File Transfer

#### 3.2.7.2 Sender procedures

##### 3.2.7.2.1 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 of 10 minutes. 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 3.2.7.1.

##### 3.2.7.2.2 Sending

When sending a RRAM to a contact, the RRAM is transported via the File Transfer service (see section 3.2.5). 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.

In complement to the procedures of section 3.2.5.3.1, 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 35.

```
<?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 35: 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 36: Example of Audio Message Transfer using File Transfer via HTTP**

#### 3.2.7.3 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 3.2.7.1, 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.

### 3.2.8 Plug-ins

#### 3.2.8.1 Overview

This section describes the procedures involved in order to share Uni-directional Plug-in content with clients that may have or may not have the Plug-in installed.

#### 3.2.8.2 Plug-in Feature tags

##### 3.2.8.2.1 Plug-in IARI value

The Plug-in IARI value is defined so that a client can declare the support for the Uni-directional plug-in framework. The value is set to *urn:urn-7:3gpp-application.ims.iari.rcs.plugin*, defined according to [3GPP TS 24.229]. An example of the IARI value carried in the IARI feature tag is shown below:

+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.plugin"

#### 3.2.8.3 Discovery

##### 3.2.8.3.1 Catalog format

The client shall populate the Plug-in entry points based on a retrieved list of available Plug-ins that is called Catalog. The Catalog is stored in the Plug-in Info server. It acts as a Plug-in whitelist and it shall correspond to the JSON Schema shown in Table 37. The Catalog may be extended further by future versions of this specification. Any extensions shall be ignored by clients that are not aware of them.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Root Object",
  "type": "object",
  "description": "Contains the plug-ins which the client can use once installed",
  "id": "http://schemas gsma.com /rcs-plugin-catalog.json",
  "properties": {
    "plug-in-catalog": {
      "title": "List of the plug-ins",
      "type": "array",
      "items": {
        "$ref": "#/definitions/plug-in"
      }
    },
    "plug-in-details": {
      "title": "Details of each plug-in",
      "type": "array",
      "items": {
        "$ref": "#/definitions/item-details"
      }
    }
  },
  "definitions": {
    "plug-in": {
      "type": "object",
      "title": "A single plug-in entry",
      "properties": {
        "plugin-id": {
          "title": "Unique ID of the plug-in",
          "type": "string"
        },
        "plugin-app-version": {
          "title": "The minimum required plug-in application version",
          "type": "integer"
        },
        "text-regexp": {
```

```
        "title": "Regular expression applied to a message",
        "type": "string"
      },
      "store-title": {
        "title": "title under the plug-in icon",
        "type": "object",
        "properties": {
          "language": {
            "$ref": "#/definitions/localized-text"
          }
        }
      },
      "store-logo": {
        "title": "image url of the plug-in icon when installed",
        "type": "string",
        "format": "uri"
      },
      "store-url": {
        "title": "deep-link url into the actual store where the plug-in can be installed from",
        "type": "string",
        "format": "uri"
      },
      "supported-chat-type": {
        "title": "declares in which contexts the plug-in shall be made available",
        "type": "string",
        "enum": [
          "1to1",
          "1to1|group",
          "group"
        ]
      }
    },
    "required": [
      "plugin-id",
      "text-regexp",
      "store-title",
      "store-logo",
      "store-url",
      "supported-chat-type"
    ]
  },
  "item-details": {
    "type": "object",
    "title": "Details of a single plug-in item",
    "properties": {
      "plugin-id": {
        "title": "Unique ID of the plug-in",
        "type": "string"
      },
      "plugin-version-name": {
        "title": "The version name of the plug-in app",
        "type": "string"
      },
      "plugin-version-number": {
        "title": "The version number of the plug-in app",
        "type": "integer"
      },
      "placeholder-title": {
```

```
        "title": "title under the plug-in icon when not installed",
        "type": "object",
        "properties": {
          "language": {
            "$ref": "#/definitions/localized-text"
          }
        }
      },
      "short-description": {
        "title": "short description of the plug-in app",
        "type": "object",
        "properties": {
          "language": {
            "$ref": "#/definitions/localized-text"
          }
        }
      },
      "description": {
        "title": "description of the plug-in",
        "type": "object",
        "properties": {
          "language": {
            "$ref": "#/definitions/localized-text"
          }
        }
      },
      "placeholder-logo": {
        "title": "image url of the plug-in icon when not yet installed",
        "type": "string",
        "format": "uri"
      },
      "date-added": {
        "title": "Timestamp of when this item has been added to the catalog ",
        "description" : "This is useful to highlight in the UI that this item is new",
        "type": "string",
        "format": "date-time"
      },
      "date-updated": {
        "title": "when this item has been updated",
        "description" : "This is useful to highlight in the UI that this item has been updated",
        "type": "string",
        "format": "date-time"
      }
    },
    "required": [
      "plugin-id",
      "plugin-version-name",
      "plugin-version-number"
    ]
  },

  "localized-text": {
    "title": "list of localized strings",
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "code": {
```

```
          "title": "language code in the [ISO 639-1] format (en,gr,etc)",
          "type": "string"
        },
        "#text": {
          "title": "actual localized text",
          "type": "string"
        }
      }
    }
  }
 }
}
```


**Table 37: Plug-in Catalog schema**

| Property | Descriptions |
|---|---|
| plugin-id | This is a globally unique identifier that is used to identify a plugin allocated by the plug-in developer. The Plug-in developer shall ensure it uses the same value whenever the plug-in is included in a Catalog following the definitions in section 3.2.8.4. |
| plugin-app-version | This is the minimum Plug-in application framework version support required by the Plug-in. The value of the client to plug-in communication version (i.e. Plug-in application version) is defined in OS specific documents. If a Plug-in requires a higher version than the one supported by the client then it shall be ignored by the client", |
| text-regexp | It is used to redirect messages to the right Plug-in (see 3.2.8.8.3) |
| store-title | It is shown when the plug-in is available and usable |
| store-logo | URL of the icon shown when the plug-in is available and usable |
| store-url | URL pointing to the actual hosting store (e.g. Google Play™) |
| supported-chat-type | It indicates the UX context in which the entry point should be shown. Can be: "1to1", "group " or "1to1\|group". |
| plugin-version-name | The user readable Plug-in version. |
| plugin-version-number | The platform specific Plug-in version. |
| placeholder-title | Optional, it is shown when the Plug-in is available but not installed |
| short-description | Optional, it is used to have a short description about the plug-in |
| description | Optional, it is used to have extra description about the plug-in |
| placeholder-logo | Optional, URL of the icon shown when the Plug-in is available but not installed. |
| date-added | It is used to highlight in the UI that this item is new |
| date-updated | It is used to highlight in the UI that this item has been updated |

**Table 38: Plug-in property descriptions**

The Catalog is retrieved based on the procedures described in 3.2.8.3.2.

An example of a Catalog is shown in Table 39.

```
{
  "plug-in-catalog": [
    {
      "plugin-id": "com.example.package#exampleplugin",
       "plugin-app-version": 1,
      "text-regexp": "(?:.*)((http|https)\\\\:\\\\/\\\\/(2-dot-files.appspot.com|files.host.com)\\\\/(\\\\S+|\\\\/))(?:.*)",
      "store-title": {
        "language": [
          {
            "code": "en",
            "#text": "Follow Me"
          },
          {
            "code": "el",
            "#text": "Ακολουθήστε με"
          }
        ]
      },
      "store-logo": "http://ext-stickers.com/plugins/android/followme/logo.png",
      "store-url": "https://play.google.com/store/apps/details?id=com.vodafone.glympseplug-in",
      "supported-chat-type": "1to1|group",
    }
  ],
  "plug-in-details" :[
    {
       "plugin-id": "com.example.package#exampleplugin",
       "plugin-version-name": "1.0.0",
       "plugin-version-number": 100,
       "placeholder-title": {
        "language": [
          {
            "code": "en",
            "#text": "Follow Me"
          },
          {
            "code": "el",
            "#text": "Ακολουθήστε με"
          }
        ]
      },
      "short-description": {

      },
      "description": {
        "language": [
          {
            "code": "en",
            "#text": "Send your location and let your friends track you until you've reached a destination, download and install the Follow Me add-on from Google Play."
          },
          {
            "code": "el",
            "#text": "Στείλε την τοποθεσία σου και άσε τους φίλους σου να δουν την διαδρομή σου μέχρι και τον προορισμό σου. Κατέβασε και εγκατάστησε το πρόσθετο Ακολούθησε με από το Google Play."
          }
        ]
      },
      "placeholder-logo": "http://ext-stickers.com/plugins/android/followme/logo.png",
```

```
    "date-added": "2014-10-01T09:30:10Z",
    "date-updated": "2016-03-01T10:30:10Z"
   }
  ]
}
```


**Table 39: Example of a Catalog**

The entity that manages the Catalog shall ensure that the plug-ins that are included in the Catalog are Uni-directional plug-ins that comply with the procedures described in this section and the respective OS specific documents. Other plug-ins may be added in future versions of this document.

#### 3.2.8.3.2 Initial Catalog retrieval and refresh

To retrieve the Catalog, the client shall

- If the CATALOG URI parameter defined in section A.1.13 and A.2.4 is configured, create a retrieval URL in accordance with the definitions of [RFC3986] with the following components:

  a) the URI scheme, the and authority and potentially the query component set to the value of the CATALOG URI configuration parameter defined in section A.1.13 and A.2.4.
  b) if the query component is not included, add a query component
  c) add the URL parameters:

  - a “plugin_version” parameter as defined in Table 40 with the value taken from the plug-in application version of the client to plug-in interface defined in the OS specific documents.
  - a “client_vendor” parameter as defined in Table 40 with the value of the vendor providing the RCS client taken from the client_vendor HTTP GET parameter included in the configuration requests
  - a “client_version” parameter as defined in Table 40 with the value of the RCS client version taken from the client_version HTTP GET parameter included in the configuration requests

  Table 40 defines the parameters used by the client to create the Catalog retrieval URL

| Parameter | Type | Value |
|---|---|---|
| plugin_version | Positive integer | Contains the Plug-in application version of the client and Plug-in communication.<br>The presence of the parameter is mandatory, if the CATALOG URI parameter is configured |

| Parameter | Type | Value |
|---|---|---|
| client_vendor | String | Contains the client_vendor HTTP GET parameter included in the configuration requests (see section 2.3.2.2)<br>Note: reserved characters in the parameter value have to be represented using percent encoding in accordance with [RFC3986].<br>The presence of the parameter is mandatory, if the CATALOG URI parameter is configured |
| client_version | String | Contains the client_version HTTP GET parameter included in the configuration requests (see section 2.3.2.2)<br>Note: reserved characters in the parameter value have to be represented using percent encoding in accordance with [RFC3986].<br>The presence of the parameter is mandatory, if the CATALOG URI parameter is configured |

**Table 40: HTTP URL parameters for initial Catalog retrieval or refresh**

Example:

If the value of the CATALOG URI parameter is:
https://plugininfoserver.serviceprovider.com
and the plugin_version is: 1
and the client_vendor is: abcd
and the client_version is: RCSAndrd-3.0
Then the client's Catalog retrieval URI is:
https://plugininfoserver.serviceprovider.com?plugin_version=1&client_vendor=abcd&client_version=RCSAndrd-3.0

- otherwise, the plug-ins list is not available to the client. The client shall not make any attempt to retrieve the Catalog and it shall not populate any plug-in entry points

If upon sending the HTTP GET request for the initial Catalog retrieval, the constructed URL (which is based on the configured CATALOG URI defined in section A.1.13 and A.2.4) does not resolve through DNS, the client shall:

- assume that the plug-ins are not available and not populate entry points
- verify availability of the constructed URL after every restart of the device or client

If the client receives in result of processing the request for the initial Catalog retrieval:

- a HTTP 503 INTERNAL SERVER ERROR with a Retry-After header then the client shall retry the initial Catalog retrieval request, the recommended value to retry will be specified in the "Retry-After" header.
- any other error, then the client shall stop the initial Catalog retrieval procedure and send another request at the next client or device restart.
- a HTTP 200 OK response, the client shall:

d) store the provided list of plug-ins

e) apply UX procedures based on the list of plug-ins;
f) store the Etag and Cache-Control directive values according to the procedures of [RFC7232];

When the validity of the Catalog expires (based on the Cache-Control directive value stored in the last retrieval), the client shall send a Catalog refresh request to the URL that is constructed as described above including the stored Etag in a *if-none-match* header according to the procedures of [RFC7232] at the earliest opportunity.

If upon sending a Catalog refresh request, the constructed URL (which is based on the configured CATALOG URI) no longer resolves through DNS, the client shall

- continue using the existing Catalog for a grace period (e.g. one week)
- verify availability of the constructed URL after every restart of the device or client.

If the client receives in result of processing the Catalog refresh request:

- a HTTP 503 INTERNAL SERVER ERROR with a Retry-After header then the client shall continue using the existing Catalog until sending the retry Catalog refresh request, the recommended value to retry will be specified in the “Retry-After” header.
- a HTTP 304 NOT_MODIFIED response, the client shall:

  g) store the new Etag and Cache-Control directive value according to the procedures of [RFC7232]
  h) continue using the existing Catalog

- any other error response, the client shall continue using the existing Catalog until sending a Catalog refresh request at the next client or device restart
- a HTTP 200 OK response, the client shall:

  i) store the new Etag and Cache-Control directive value according to the procedures of [RFC7232]
  j) store the provided Catalog
  k) apply UX procedures based on the received Catalog

When data is switched off (see section 2.8.1.5), this interface shall remain available if device management is configured as a cellular data off exempted service via the client configuration parameters Device Management over PS data off exemption or Device Management over PS data off roaming exemption defined in section A.1.14. If that is not the case and the client shall send a request for initial Catalog retrieval or Catalog refresh, the client shall send the request at the earliest opportunity (e.g. when data is switched on again or when connecting over Wi-Fi).

For the Expires header, similarly to section 3.6.4.1.2, the Service Provider shall ensure that the entity that maintains the Catalog returns responses with Expires header value set to a date in the past. This will ensure that legacy proxies do not attempt to cache the content.

#### 3.2.8.4 Plugin-id

A globally unique identifier that is used to identify a Plug-in allocated by the plug-in developer. The Plug-in developer shall ensure it uses the same value whenever the plug-in is included in a Catalog. It is defined in ABNF as follows:

```
plugin_id = top-level-domain "." organization "." pluginidentifier
pluginidentifier = appidentifier ["#" pluginitem]

top-level-domain = "com" / "edu" / "gov" / "mil" / "net" / "org" /
iso3166countrycode

organization = identifier
appidentifier = identifier
pluginitem = identifier
identifier = ALPHA *(ALPHA / DIGIT)
iso3166countrycode = 2ALPHA; country code as in ISO31666
```

The value shall not contain the character which is used as a separator in the new CPIM header i.e. the "_" (see 3.2.8.6).

The Plugin-id is case insensitive.

#### 3.2.8.5 Privacy Protection

##### 3.2.8.5.1 Plug-in Authorization

The Plug-ins that are included in the Catalog are the Plug-ins that are authorized to be used by the Service provider.

If the Service provider wants to block a Plug-in, they shall remove the Plug-in from the Catalog. The Plug-in will be removed from the client in the next Catalog refresh request.

#### 3.2.8.6 Traffic identification

##### 3.2.8.6.1 CPIM Namespace

The new CPIM namespace defined for the Messaging as a Platform (MaaP) related CPIM headers (see section 3.6.7) shall be used.

##### 3.2.8.6.2 New CPIM header Plugin-Info

The header is defined as an extension to the [RFC3862] field definitions. The limits for the occurrence of the field are defined in the following table:

| Field | Min Number | Max Number |
|---|---|---|
| Plugin-Info | 0 | 1 |

**Table 41: Plugin-Info header**

The field itself is defined in ABNF as follows:

```
plugin-info = "Plugin-Info" ":" SP plugin-info-value CRLF
plugin-info-value = plugin-id [ "_" plugin-content-id ]
plugin-id = 1*allowed-chars
plugin-content-id =1*allowed-chars
allowed-chars = ALPHA / DIGIT / "."
```

Example:

```
maap.Plugin-Info: com.gsma.pluginsample_ab4
```

The plugin-id is mandatory and the client gets the value from the Catalog (see 3.2.8.3.1) or the Plug-in to Client interactions (see 3.2.8.7.1). Its value shall follow the definitions of section 3.2.8.4. The plugin-content-id is optional and the client gets it from the Plug-in to Client interactions (see 3.2.8.7.1). The plugin-content-id shall be included for Plug-ins that generate replicable content (e.g. stickers).

#### 3.2.8.7 Sender procedures

##### 3.2.8.7.1 Plug-in to client interaction

The client shall request the Plug-in to generate the Plug-in content or the link that points to the content (depending on the Plug-in). Upon content generation request, the client shall indicate to the Plug-in the list of possible content types i.e. mime-types (see Table 42) for the Plug-in content based on:

- the Plug-in supported mime-types declared to the client and
- for one to one conversations, the capabilities supported by the receiver

The operations between the client and the plug-in for generating the Plug-in content or the link that points to the content are OS specific and are described in the relevant documents.

##### 3.2.8.7.2 Service selection and delivery procedures

The Uni-directional plug-in content (or link to the content) is transferred using the Messaging services (see section 3.2).

The Messaging service selection procedures are based on the conversation type i.e. 1-to-1 conversations or Group conversations the plug-in content is transferred:

- For 1-to-1 conversations, the client provides the list of the supported mime-types to the plug-in based on the capabilities of the receiver. The plug-in selects the mime-type of the plug-in content taking into account the list of supported mime-types provided by the client. The client selects the Messaging service based on the received plug-in mime-type. There is no impact on the selected Messaging service based on the presence of the Plug-ins capability (see section 2.6.1.3).
- For group conversations, the client provides the list of supported mime-types to the plug-in based on the capabilities supported in the group conversation. The plug-in selects the mime-type of the plug-in content taking into account the list of supported mime-types provided by the client. The client selects the Messaging service based on the received plug-in mime-type.

| Plug-in mime-type | Selected Messaging service | |
|---|---|---|
| | 1-to-1 conversation | Group Conversation |
| text/plain | 1-to-1 Messaging (see 3.2.1) | Group Chat (see 3.2.4) |
| image/* or video/mpeg4 | Sending File Transfer to a single user (see 3.2.5) | Sending File Transfer to multiple users (see 3.2.5) |
| audio/mp3 | Sending Audio message to a single user (see 3.2.7) | Sending Audio message to multiple users (see 3.2.7) |

| Plug-in mime-type | Selected Messaging service | |
|---|---|---|
| | 1-to-1 conversation | Group Conversation |
| application/vnd.gsma.rcspushlocation+xml | Sending Geolocation Push to a single user (see 3.2.6) | Sending Geolocation Push to multiple users (see 3.2.6) |

**Table 42: Messaging service selection based on Plug-in mime-type**

The delivery procedures of the underlying selected Messaging service apply.

When data is switched off (see section 2.8.1.5) Uni-directional Plug-ins shall be available if according to the configuration parameter of the underlying selected Messaging service defined in section A.1.14 (i.e. RCS MESSAGING DATA OFF or FILE TRANSFER DATA OFF), the service should remain available.

#### 3.2.8.7.3 Sending Uni-directional Plug-in content for 1-to-1 conversations

Based on the selected Messaging service and technology, the client shall follow the procedures as described in the respective sections of this specification.

The CPIM messages shall include the CPIM header defined in section 3.2.8.6.2.

#### 3.2.8.7.4 Sending Uni-directional Plug-in content for Group conversations

Based on the selected Messaging service, the client shall follow the procedures as described in the respective sections of this specification.

The CPIM messages shall include the CPIM header defined in section 3.2.8.6.2.

### 3.2.8.8 Receiver procedures

#### 3.2.8.8.1 Receiving Uni-directional Plug-in content for 1-to-1 conversations

Based on the selected Messaging service and technology, the client shall follow the procedures as described in the respective sections of this specification.

#### 3.2.8.8.2 Receiving Uni-directional Plug-in content for Group conversations

Based on the selected Messaging service, the client shall follow the procedures as described in the respective sections of this specification.

NOTE: If the plug-in generates text/plain content that carries a link, the download procedures for content localisation described in section 3.2.5.3.2.1 will not apply.

#### 3.2.8.8.3 Client to Plug-in interaction

The client shall identify the Plug-in generated content by checking:

- In case of CPIM messages, the Plugin-Info value, when the Plugin-Info CPIM header (defined in 3.2.8.6.2) is present
- In case of CPIM messages, if the text matches any regular expression of the enumerated installed Plug-ins when the new Plug-in Info CPIM header is not present.
- In case of SMS, if the text matches any regular expression of the enumerated installed Plug-ins.

- In case of SMS, if the text matches any regular expression of the plug-ins listed in the Catalog.

If the identified Plug-in is installed, the client shall direct the content to the Plug-in and request the Plug-in to display it. The operations between the client and the Plug-in for displaying the Plug-in content are OS specific and are described in the relevant documents.

If the identified Plug-in is available but not installed then the client shall apply the UX procedures and offer an entry point to the store page where the Plug-in can be downloaded. The store location is retrieved from the Catalog.

#### 3.2.8.8.4 New CPIM header Plug-Info towards legacy clients

To avoid issues with the handling of the new CPIM Plug-Info header in older version RCS clients, the Messaging Server serving such a client shall remove the CPIM Plug-Info header and conditionally the corresponding CPIM NS header prior to message delivery via the procedures defined in section 2.14.

## 3.3 Content sharing

### 3.3.1 In-Call services

#### 3.3.1.1 Void

#### 3.3.1.2 Shared Map

The technical realisation is based on procedures covered in section 2.9.7 and 2.9.9 and 2.9.10 of [PRD-RCC.20].

#### 3.3.1.3 Shared Sketch

The technical realisation is based on procedures covered in section 2.9.8, 2.9.9 and 2.9.10 of [PRD-RCC.20].

#### 3.3.1.4 Interaction of In-Call services with voice Call

The Shared Map and Shared Sketch services during a voice call (either over CS or as specified in section 3.4) 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.

NOTE: This interaction does not apply for the File Transfer and 1-to-1 chat service. The sharing session is independent of that voice call and progresses independently of the voice call continuity.

##### 3.3.1.4.1 Multiparty call and In-Call sharing services

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, the Shared Map and Shared Sketch 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 Shared Map or Shared Sketch 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 2.6.1) indicating that the Content Sharing services during a call are no longer available. The on-screen icons/layout should be updated accordingly.

- The users (User A and User B) were in an active Shared Map session: In this case, switching to a multiparty call means ending the Shared Map session. This can be initiated by either user (user A or user B) depending upon the circumstances. 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 users (User A and User B) were in an active Shared Sketch session: In this case, switching to a multiparty call means ending the Shared Sketch session. This can be initiated by either user (user A or user B) depending upon the circumstances. 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.

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 2.6.1.

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.

#### 3.3.1.4.2 Call on hold and In-Call sharing services

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, the Shared Map and Shared Sketch 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 Shared Map or Shared Sketch 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 2.6.1) indicating that the Content Sharing services during a call are no longer available. The on-screen icons/layout should be updated accordingly.
- The users (User A and User B) were on an active Shared Map session: In this case, putting the call on hold means ending the Shared Map session. This can be initiated by either user (User A or User B) depending upon the circumstances. 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 users (User A and User B) were on an active Shared Sketch session: In this case, putting the call on hold means ending the Shared Sketch session. This can be initiated by either user (user A or user B) depending upon the circumstances. 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.

It should also be 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 2.6.1, and,
- If there is another active call, the behaviour regarding the Shared Map and Shared Sketch 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.

#### 3.3.1.4.3 Waiting call and In-Call sharing services

A waiting call is a non-active call; therefore, it should not be possible to access the Shared Map and Shared Sketch services between the caller and receiver.

Please note having a waiting call will not affect the behaviour for Shared Map and Shared Sketch services (that is both for the capability exchange and the services itself) on the active call.

#### 3.3.1.4.4 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 Shared Map and Shared Sketch services between the caller and receiver.

#### 3.3.1.4.5 Call divert/forwarding

A receiver may have call divert/forwarding active (the calls are for instance forwarded to another number or to voicemail), it is still possible to access the Video Share, Shared Map or Shared Sketch services from the caller to the receiver if, as per section 7.3.1.2 of [3GPP TS 24.279]:

- The caller has received a P-Asserted-Identity value from the receiver, or
- The caller has received a Connected Number information element and implements the procedure from section 7.3.1.2 of [3GPP TS 24.279].

Otherwise, it is not possible to access the Shared Map and Shared Sketch services from the caller to the receiver.

### 3.3.2 Other Content Sharing Services

#### 3.3.2.1 Call composer

The technical realisation for both Call Composer via Enriched Calling session and Call Composer via Multimedia Telephony session is based on procedures covered in sections 2.3 and 2.4 of [PRD-RCC.20].

#### 3.3.2.2 Post-call service

The technical realisation is based on procedures covered in sections 2.3 and 2.5 of [PRD-RCC.20].

#### 3.3.2.3 Call Composer flows

Flows related to the two services Call Composer via Enriched Calling session and Call Composer via Multimedia Telephony session are provided in Annex A of [PRD-RCC.20].

## 3.4 IP Voice Call

### 3.4.1 Overview

At a technical level the voice call service shall be based on [PRD-IR.92] and [PRD-IR.51] and may either be realised

- As described in [PRD-IR.92] and [PRD-IR.51] for primary devices enabled for VoLTE/VoWiFi in which case [PRD-NG.102] shall be supported as well. Or
- As described in section 3.4.2 for other devices that do not support CS voice calls (i.e. secondary devices).

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.51], the support for early media as described in [PRD-IR.92] and [PRD-IR.51] is required.

Broadband Access clients which support and are configured for RCS IP Voice Call but are not enabled for VoLTE/VoWiFi (and therefore do not make use of the IMS APN as specified in section 2.8.1.4) shall behave as defined in section 3.4.2.

NOTE: When using the RCS IP Voice Call service, it is recommended that the device indicate to the user that this is not a telephony replacement service.

### 3.4.2 Devices using RCS IP voice calls

Table 43 summarises the sections in [PRD-IR.92] and [PRD-IR.51] that apply and do not apply to an RCS IP Voice 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.51] | Applicability |
|---|---|---|
| **[PRD-IR.92], [PRD-IR.51]** | **All sections not mentioned below** | **Applicable** |

| Document(s) | Relevant sections from [PRD-IR.92], [PRD-IR.51] | Applicability |
|---|---|---|
| [PRD-IR.92], [PRD-IR.51] | 2.2.1 SIP Registration Procedures | The MMTEL IMS Communication Service Identifier (ICSI) shall be included in the Contact header field.<br>According to the rules defined in section 2.4.3 *+g.gsma.rcs.telephony* is either also included in the Contact header field and filled with the values *cs* or *none* or it is not included in the *Contact* header field at all.<br>In addition if RCS IP Voice Call is enabled but RCS IP Video Call is not enabled the client shall also include *+g.gsma.rcs.ipcall* in the *Contact* header field<br>See also section 2.4 |
| [PRD-IR.92], [PRD-IR.51] | 2.2.2 Authentication | Not applicable. See section 2.12 |
| [PRD-IR.92], [PRD-IR.51], | 2.2.4 Call Establishment and Termination | 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].<br><br>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]. |
| [PRD-IR.51] | 2.2.7 Hosted NAT Traversal | Not applicable. See section 2.7 |
| [PRD-IR.92], [PRD-IR.51] | 2.4.2 Integration of resource management and SIP | Applicable only if the IMS well-known APN is used for RCS (see section 2.8.1.4) |
| [PRD-IR.92], [PRD-IR.51] | 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.51] | 4 Radio and packet core feature set | Entire section (including subsections) not applicable |
| [PRD-IR.92] | 4.1 Robust Header Compression | Not applicable |
| [PRD-IR.92] | 4.2 LTE Radio Capabilities | Not Applicable |
| [PRD-IR.92] | 4.3 Bearer Management | Not Applicable |
| [PRD-IR.92] | 4.4 P-CSCF Discovery | Not Applicable, see section 2.4.5 |
| [PRD-IR.92], [PRD-IR.51] | 5.1 IP Version | Not Applicable |
| [PRD-IR.92] | 5.2 Emergency Service | Subject to local regulation |
| [PRD-IR.51] | 5.3 Emergency Service | Subject to local regulation |

| Document(s) | Relevant sections from [PRD-IR.92], [PRD-IR.51] | Applicability |
|---|---|---|
| [PRD-IR.92] | 5.3 Roaming Considerations | Not Applicable |
| [PRD-IR.51] | 5.4 Roaming Considerations | Not Applicable |
| [PRD-IR.92] | 5.4 Accesses in addition to E-UTRAN | Not Applicable |
| [PRD-IR.92] | 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)<br>In [PRD-IR.94] Annex A Complementing IMS with CS (A.1 General, A.2 SR-VCC) | Not Applicable |

**Table 43: IR.92 and IR.51 applicability to RCS IP Voice Call**

A device/client supporting and configured to use RCS IP Voice Calls shall indicate this in SIP INVITE requests and responses according to Table 43. 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
```

## 3.5 IP Video Call

### 3.5.1 Overview

Depending on the client configuration and client limitations, three technical are available to provide access to the IP Video Call service:

- ViLTE (Video over LTE) as defined in [PRD-IR.94],
- ViWiFi (Video over WiFi) as defined in [PRD-IR.51] for EPC integrated WiFi access and
- RCS IP Video Call as described in section 3.5.2 for generic data access.

A device enabled for VoLTE/VoWiFi shall offer IP Video Call only according to [PRD-IR.94] and [PRD-IR.51]. Integration of resource management and SIP is done as per [PRD-IR.94] for devices currently supporting VoLTE, and as per [PRD-IR.94] and [PRD-IR.51] for devices currently supporting VoWiFi.

For other devices (i.e. primary devices not enabled for VoLTE/VoWiFi or secondary devices), the device shall offer IP Video Call as an RCS IP Video Call according to section 3.5.2. This would mean providing this service on a best effort basis. For such devices, the service will only be available depending on the Service Provider policy settings (*PROVIDE RCS IP VIDEO CALL* as defined in section A.1.11). For such devices, no specific requirements for resource management are required.

The technical enabler used for IP Video Call is not required to be the same on both sides since they are fully compatible.

For RTP media and RTCP usage, a device using non-cellular access shall follow the requirements for NAT traversal as specified in section 2.7.

### 3.5.2 Devices using RCS IP video calls

Table 44 summarizes the sections in [PRD-IR.92] and [PRD-IR.51] 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.51], [PRD-IR.94] | Applicability |
|---|---|---|
| **[PRD-IR.94]** | **All sections not mentioned below** | **Applicable** |
| **[PRD-IR.92], [PRD-IR.51]** | **All sections not mentioned below that are referred to from [PRD-IR.94]** | **Applicable** |
| [PRD-IR.92], [PRD-IR.51], [PRD-IR.94] | 2.2.1 SIP Registration Procedures | The MMTEL IMS Communication Service Identifier (ICSI) shall be included in the Contact header field.<br>According to the rules defined in section 2.4.3 *+g.gsma.rcs.telephony* is either also included in the Contact header field and filled with the values *cs* or *none* or it is not included in the *Contact* header field at all.<br>In addition,<br>• If RCS IP Video Call is enabled the client shall also include *+g.gsma.rcs.ipcall;video* in the *Contact* header field.<br>• 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 3.5.2.1) the client shall also include *+g.gsma.rcs.ipcall;+g.gsma.rcs.ipvideocallonly;video* in the *Contact* header field.<br>See also section 2.4 |
| [PRD-IR.92], [PRD-IR.51] | 2.2.2 Authentication | Not applicable. See section 2.12 |

| Document(s) | Relevant sections from [PRD-IR.92], [PRD-IR.51], [PRD-IR.94] | Applicability |
|---|---|---|
| [PRD-IR.92], [PRD-IR.51], [PRD-IR.94] | 2.2.4 Call Establishment and Termination<br>(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].<br><br>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].<br><br>In addition to the above,<br>• For an RCS IP Video Call the client shall also include *+g.gsma.rcs.ipcall;video* in the Contact header field.<br>• 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.ipvideocallonly;video* in the *Contact* header field. |
| [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] | 3.2.6 Jitter Buffer Management Considerations | Not applicable |
| [PRD-IR.92] | 3.2.7 Front End Handling | Not applicable |
| [PRD-IR.51] | 4 Radio and packet core feature set | Entire section (including subsections) not applicable |
| [PRD-IR.92] | 4.1 Robust Header Compression | Not applicable |
| [PRD-IR.94] | 4.2 Bearer Considerations for Video | Not Applicable |
| [PRD-IR.94] | 4.3 LTE Radio Capabilities | Not Applicable |
| [PRD-IR.92] | 4.4 P-CSCF Discovery | Not Applicable, see section 2.4.5 |
| [PRD-IR.92], [PRD-IR.51] | 5.1 IP Version | Not Applicable |
| [PRD-IR.92] | 5.2 Emergency Service | Subject to local regulation |
| [PRD-IR.51] | 5.3 Emergency Service | Subject to local regulation |

| Document(s) | Relevant sections from [PRD-IR.92], [PRD-IR.51], [PRD-IR.94] | Applicability |
|---|---|---|
| [PRD-IR.92] | 5.3 Roaming Considerations | Not Applicable |
| [PRD-IR.51] | 5.4 Roaming Considerations | Not Applicable |
| [PRD-IR.92] | 5.4 Accesses in addition to E-UTRAN | Not Applicable |

**Table 44: IR.92, IR.94 and IR.51 applicability to RCS IP Video Call**

A device/client supporting and configured to use RCS IP video calls shall indicate this according to Table 44.

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
```

#### 3.5.2.1 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.1.11) 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 header field 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 header field 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 Contact header field 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.

## 3.6 Chatbots

This section describes the architecture and technical enablers supporting Chatbots in 1-to-1 conversations to realise the functionality described in section 15.2 of [PRD-RCC.71].

### 3.6.1 Architecture

Chatbots require support functionality in both the RCS Service Provider network and the Chatbot Platform. Figure 13 provides an overview of how that functional split is done whereby optional functions in Chatbot Platform or Service Provider network are indicated through dashed lines shaded boxes. Interfaces that are out of scope of this specification are indicated in a similar way.

NOTE: In this section, it has been assumed that the Chatbot Platform has been integrated in the overall architecture for RCS as described in section 2.1 and [PRD-IR.90]. The section therefore focuses on the mentioned functional split between the Chatbot Platform and the Service Provider network assuming that the connection between these is in place.

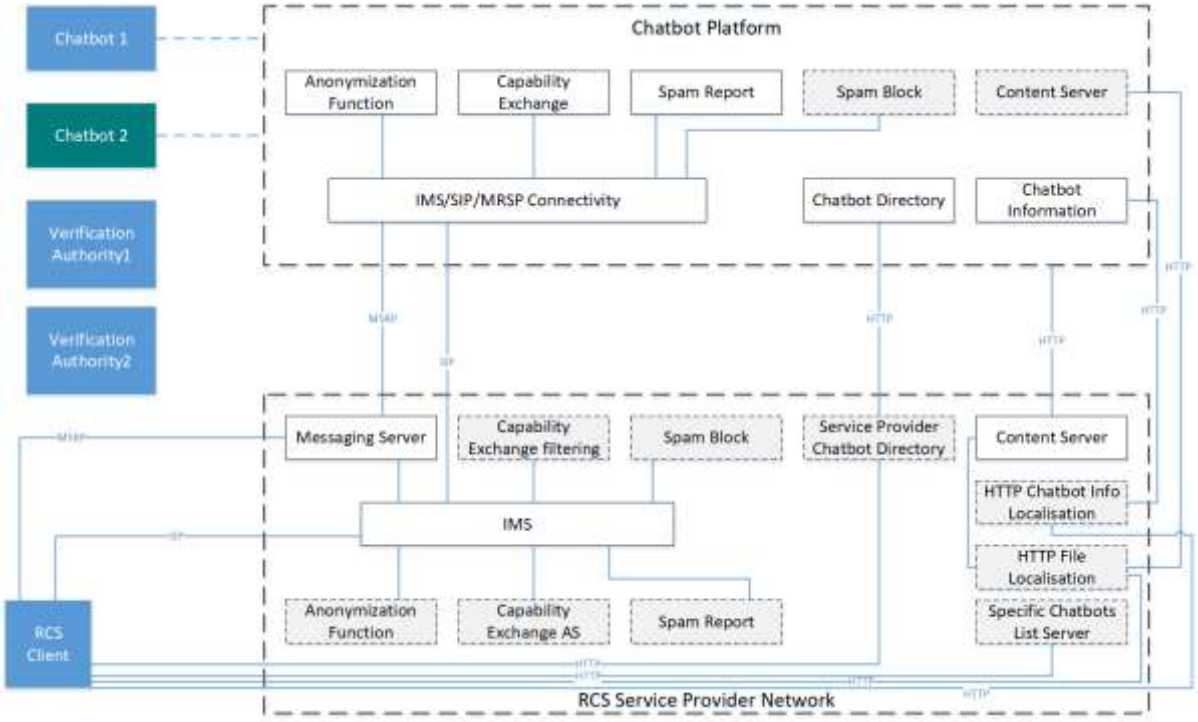



**Figure 13: Functional Split between Service Provider Network and Chatbot Platform**

NOTE: Figure 13 shows the case where HTTP Localisation functions are present. If not configured to use HTTP localisation for a function, the client shall contact the corresponding function directly.

NOTE: If the Chatbot Platform is provided by the Service Provider themselves, the Chatbot Platform is network internal and therefore it is up to the Service Provider to define the functional split between the different functions in their network.

NOTE: Figure 13 is a functional split, which is also reflected in the naming of the functions (e.g. Service Provider Chatbot Directory). Depending on commercial agreements and chosen deployment model, functions in the RCS Service Provider Network could be provided also by the Chatbot Platform or by a 3rd party provider. Similarly, functions in the Chatbot

Platform could also be provided by an RCS Service Provider or by a 3rd party provider.

In the Service Provider network, the following functionality shall be required to enable the Chatbot use cases:

- SIP/IMS routing and authentication functionality to receive the corresponding requests and responses from the client and the Chatbot Platform and route them to the required functions in the network
- A Messaging Server supporting 1-to-1 conversations with deferred messaging functionality
- Functionality to provide interworking to xMS is not required
- A function to support Capability Exchange that is mandatory when the interconnection to the Chatbot Platform relies on SIMPLE Presence for the Capability Exchange (see section 2.6.1.2) and optional if SIP OPTIONS is used (see section 2.6.1.1.1).
- An optional Spam Report Function which is required only if the RCS Service Provider wants to process the Spam Reports sent by their subscribers (see section 3.6.6.2)
- An optional Spam Block Function preventing bots considered as Spam senders from contacting and being contacted by RCS users (see section 3.6.6).
- An optional Specific Chatbots List Server that may be providing the client with lists of Chatbot Service IDs (see section 2.5.4.1) of Chatbots requiring specific management (see section 3.6.3.3).
- An optional Anonymization Function as described in section 3.6.5.1 which is required if subject to the interworking agreements for the connection with the Chatbot Platform anonymization of the user towards the Chatbot is done by the Service Provider
- An optional Capability Exchange Filter Function for the filtering of capabilities exchanged with the Chatbot Platform if the RCS Service Provider wants to limit the RCS Services available to the Chatbot Platform as described in section 15.2.1 of [PRD-RCC.71]
- A Content Server to host the file content sent by the user to the Chatbot Platform
- An optional Service Provider Chatbot Directory Function to allow the user to discover Chatbots as described in section 3.6.3.1.

NOTE: The interfaces and processes to populate the directory are out of scope of this specification.

- Optional HTTP Localisation Functions to ensure that the HTTP traffic relating to Chatbots is routed through the Service Provider Network (as described in section 3.2.5.7.4). If this is not provided the client shall interact directly with the Chatbot functions that rely on HTTP (i.e. Content Servers and Chatbot Information Function).

In the Chatbot Platform, the following functionality shall be required to enable the Chatbot use cases:

- Connectivity for SIP/MSRP to send and receive requests from the RCS Service Providers relating to the Chatbots that it hosts

NOTE: In future, other protocols may be defined to provide this functionality.

- An Anonymization Function which is required if subject to the interworking agreements for the connection with the Chatbot Platform anonymising the user identity towards the Chatbot is done by the Chatbot Platform
- Functionality to answer Capability Exchange requests for a Chatbot's capabilities
- Functionality to receive Spam Reports
- An optional Spam Block Function preventing bots considered as Spam senders from contacting RCS users
- An optional Content Server to host content sent by the Chatbots to the user. If this is not provided, the content exchanged must be hosted on webservers available on the internet and would be accessible only by users of RCS Service Providers that allow access to such content.
- A Chatbot Information Function providing contact details and other information on the Chatbots hosted by the Chatbot Platform as described in section 3.6.4.
- A Chatbot Directory Function to allow the RCS Service Provider to retrieve the Chatbot Information of the hosted Chatbots.

Next to those, there are Verification Authorities that act as a 3rd party verifying the data provided by the Chatbot in the Chatbot Information.

The following is not in scope of this specification:

- The interfaces that are internal to a Service Provider Network or Chatbot Platform. This includes the interfaces between the Service Provider Network and the Chatbot Platform for the case where both are provided by the same Service Provider.
- The interfaces between Chatbots and Verification Authorities and the interfaces between Verification Authorities and RCS Service Provider networks. Those interfaces are based on business processes rather than technical procedures.
- The interface between Chatbot and Chatbot Platform
- The Chatbots' logic
- The interaction of Chatbots with xMS messaging
- The Service Provider Chatbot Directory Function logic i.e. which entries will be returned based on the provided parameters and the information obtained from the Chatbot Platform Directory Function and in which order those entries will be returned

### 3.6.2 Chatbot Feature tags

#### 3.6.2.1 Chatbot IARI value

The Chatbot IARI value is defined so that an entity can declare its support for Chatbots in general.

If the Chatbot communication is based on Chatbot Chat Sessions, the Chatbot IARI value is set to *urn:urn-7:3gpp-application.ims.iari.rcs.chatbot*, defined according to [3GPP TS 24.229]. An example of the IARI value carried in the IARI feature tag is shown below:

+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.chatbot"

If the Chatbot communication is based on Chatbot Standalone Messages, the Chatbot IARI value is set to *urn:urn-7:3gpp-application.ims.iari.rcs.chatbot.sa*, defined according to [3GPP TS 24.229]. An example of the IARI value carried in the IARI feature tag is shown below:

+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.chatbot.sa"

In the capability exchange (see section 2.6.1.3), if both Message technologies are supported, the two IARIs shall be carried together. An example is shown below:

+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.chatbot,urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.chatbot.sa"

#### 3.6.2.2 Chatbot application version

The Chatbot application version is represented by a decimal number. It indicates support by the entity for the content-types and procedures defined in this specification for Chatbot communication. The Chatbot application version shall be increased with any new release by "1". The initial version of the Chatbot application version is "1". It indicates to an entity the Chatbot application versions supported by the entity to which it has connected.

The Chatbot application version defined in this document is "1".

The indication of the Chatbot application version supported by an entity uses a feature parameter as defined in [RFC3840]. The feature parameter is encoded as an "other-tags" feature tag in accordance with the definitions in section 9 of [RFC3840].

The feature tag name shall be set to "g.gsma.rcs.botversion". The tag value shall be encoded as "numeric" in accordance with the definitions in section 9 of [RFC3840]. Multiple tag values can be included in the Chatbot application version media feature tag.

Security considerations for this media feature tag are discussed in section 11.1 of [RFC3840].

Non-normative examples:

- A Chatbot application version media feature tag indicating the support of all Chatbot application versions up to "4" could be represented as follows:
  - +g.gsma.rcs.botversion="#<=4", or
  - +g.gsma.rcs.botversion="#=1,#=2,#=3,#=4"
- A Chatbot application version media feature tag indicating the support of Chatbot application versions "2", "4", "5" and "6" could be represented as follows:
  - +g.gsma.rcs.botversion="#=2,#=4,#=5,#=6", or
  - +g.gsma.rcs.botversion="#=2,#4:6"

#### 3.6.2.3 Chatbot role

The indication of the Chatbot role is a boolean expression with a value of "true". It indicates to a client that the server to which it has connected is representing a Chatbot.

The indication of the Chatbot role is provided by a feature parameter as defined in section 9 of [RFC3840]. The feature parameter is encoded as an "other-tags" feature tag in accordance with the definitions of section 9 of [RFC3840].

The feature tag name shall be set to "g.gsma.rcs.isbot". The media feature tag shall have no value.

Security considerations for this media feature tag are discussed in section 11.1 of [RFC3840].

The indication of the Chatbot role is represented as follows:

+g.gsma.rcs.isbot

#### 3.6.2.4 Chatbot capability

A client shall consider that a contact is a Chatbot if the SIP URI of the contact is without a user=phone parameter and

- the domain part of the SIP URI of the contact includes the “botplatform” subdomain as shown in Table 5, or
- where based on a capability exchange or message exchange an indication was provided that the Contact is a Chatbot (i.e. the Chatbot role as defined in Table 9 was provided), or
- where an earlier messaging communication request has been referred to a Chatbot messaging communication via the procedures in section 3.2.1.1.

### 3.6.3 Discovery and specific management of Chatbots

#### 3.6.3.1 Client to Service Provider Chatbot Directory interface

A Service Provider Chatbot Directory is a logical entity in the RCS Service Provider network where a list of available Chatbots, aggregated from the interconnected Chatbot Platforms, is maintained based on which Chatbots the Service Provider wishes to make discoverable by its RCS users. This list can be retrieved, parsed and displayed with their metadata.

If the RCS service provider wishes to provide client access to the Service Provider Chatbot Directory, they shall configure a Chatbot Directory URL from where the client can retrieve a list of Chatbots through the CHATBOT DIRECTORY client configuration parameter defined in section A.1.3. If the client configuration parameter is not provided, the client shall assume that the Service Provider Chatbot Directory functionality is disabled. If the URL derived from the CHATBOT DIRECTORY client configuration parameter cannot be resolved through DNS, the client shall

- assume that the Service Provider Chatbot Directory functionality from the service provider is not available and
- verify availability of the configured URL after every restart of the device or every restart of the RCS client.

If the Chatbot Directory URL is available and accessible, the client shall send a HTTP GET request to retrieve the referred resource including the query parameters defined in Table 45 when Chatbot discovery is invoked through UX. If the provisioned Chatbot Directory URL

includes query parameters already, the client shall include the parameters defined in Table 45 in the existing query component. Otherwise, the client shall add a query component to the URL and include the parameters defined in Table 45 in that query component. The client implementation shall display the Chatbots in the same order that the Chatbot entries are ordered in the response that is provided on this HTTP GET request. Caching of this response may be implemented on the client for efficiency e.g. when the same query is triggered by the UX within some seconds. It is left to the client implementation to set the validity time of this cached response and to apply further optimisations.

| Query Parameter | Description | Mandatory | Format |
|---|---|---|---|
| q | Query terms that can specify words or phrases to filter the query results that the Service Provider Chatbot Directory returns. The value must be URL-escaped. | N | String |
| start | The start parameter indicates the first matching result that should be included in the query results. It uses a zero-based index, meaning the first result is 0; the second result is 1 and so forth. The start parameter works in conjunction with the num parameter to determine which query results to return. | N | Positive integer |
| num | The num parameter identifies the number of query objects to return. If not provided, the num value is determined by the server. | N | Positive integer |
| pl | It indicates the preferred language(s) for Chatbot search. Its default value is the selected device language setting. The default value may be edited by the user or additional values may be added based on user preferences. Its format shall follow the [ISO 639-1] definitions. It is provided based on user setting to enrich the search.<br>Example:<br>If the selected device setting is English and the user additional preferred languages for search is Spanish and German then the “pl” parameter is as follows in the request:<br>pl=en&pl=es&pl=de | N | String<br>Multi-valued parameter |
| lat | It indicates the latitude of the device/client based on signed degrees’ format. It is provided based on user setting to enrich the search. | N | Double |
| lon | It indicates the longitude of the device/client based on signed degrees’ format. It is provided based on user setting to enrich the search. | N | Double |
| ho | It indicates the home operator of the device and this shall be represented as <MCCMNC> combination. Whereby MCC and MNC 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. | Y | String |

| Query Parameter | Description | Mandatory | Format |
|---|---|---|---|
| i | It indicates the MSISDN of the user. The client shall provide the MSISDN value with the plus sign encoded as per [RFC3986].It is provided based on user setting to enrich the search and the value of the IDENTITY IN ENRICHED SEARCH parameter defined in section A.1.3 and A.2.4. | N | E.164 international format |
| chatbot_version | It contains the Chatbot application version the client supports defined in 3.6.2.2. If not provided, then it means only version 1 is supported.<br>The version parameter shall be composed as follows:<br>version = version_number * [ "_" version_number ]<br>version_number = 1*DIGIT<br>Example: chatbot_version=1_2_3 | N | String |
| client_vendor | It contains the query client_vendor HTTP GET parameter included in the configuration requests (see 2.3.2.2). | Y | String |
| client_version | It contains the query client_version HTTP GET parameter included in the configuration requests (see 2.3.2.2). | Y | String |

**Table 45: Query parameters for the Client to Directory query**

How these query parameters included in the URL are handled and the query results in the response are prepared is up to the Service Provider Chatbot Directory function implementation and is out of the scope of this specification. If query parameters are provided that are not known by the Service Provider Chatbot Directory (e.g. because the querying client complies with a future version of this specification), it is up to the Service Provider Chatbot Directory function implementation how to handle them.

The Service Provider Chatbot Directory shall return the query result in the JSON format with a HTTP 200 OK response. The schema in Table 46 defines all JSON payloads exchanged between the Service Provider Chatbot Directory and the query client.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
    "properties": {
      "bots": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "id": {
              "type": "string",
              "format": "uri"
            },
            "name": {
                "type": "string"
            },
            "icon": {
```

```
            "type": "string",
            "format": "uri"
          },
          "verified": {
            "type": "boolean"
          }
        },
        "required": ["id","name"]
      }
    },
    "itemsReturned": {
      "type": "integer"
    },
    "startIndex": {
      "type": "integer"
    },
    "totalItems": {
      "type": "integer"
    }
  },
  "required": ["bots"]
}
```

**Table 46: JSON Schema for Service Provider Chatbot Directory payloads**

A detailed description of each JSON object is specified in Table 47. The client shall gracefully ignore any JSON object that it does not recognize.

| JSON | Description |
|---|---|
| id | Chatbot service ID (see NOTE below) |
| name | Chatbot name |
| icon | Link to the Chatbot icon image. If not provided, the client shall assume that the icon is not available. |
| verified | Indicates whether the Chatbot is to be presented to the user as a Chatbot whose identity has been verified. If absent, the Chatbot shall not be presented as verified. |
| itemsReturned | The number of query objects returned |
| startIndex | The index of the first query result in the current set of query results |
| totalItems | The number of query objects available |

**Table 47: Description of the Service Provider Chatbot Directory query result**

NOTE: An id value can occur only once in the query results. If there are more than one entry available with the same Chatbot service ID, it is up to the Service Provider Chatbot Directory Function to select which one to return to the client.

#### 3.6.3.2 Directory to Directory interface

To allow the Service Provider Chatbot Directory function to aggregate the list of hosted Chatbots of the Chatbot Platforms that the RCS Service provider is interconnected with, the Service Provider Chatbot Directory function can send a HTTP GET request to retrieve the Chatbot information details towards the Chatbot Directory of the interconnected Chatbot Platforms. The Chatbot Directory shall respond based on the interconnection agreement

between the RCS service provider and the Chatbot Platform. The frequency with which this HTTP GET request is sent is based on the RCS Service Provider policies and it may be subject to the interconnection agreement between the RCS Service Provider and the Chatbot Platform. That agreement is outside of the scope of this specification. The Service Provider Chatbot Directory shall send a HTTP GET request to retrieve the searchable Chatbot information for the different Chatbots with the query parameters defined in Table 48.

| Query Parameter | Description | Mandatory | Format |
|---|---|---|---|
| qo | It indicates the querying operator and it shall be represented as <MCCMNC> combination. Whereby MCC and MNC shall be replaced by the respective values of the querying operator in decimal format and with a 2-digit MNC padded out to 3 digits by inserting a 0 at the beginning. It is provided based on Service Provider Chatbot function policies. | N | String<br>Multi-valued parameter |
| chatbot_version | It indicates the Chatbot application version of the clients that the Service Provider Chatbot Directory server supports. It is defined in section 3.6.2.2. If not provided, then it means only version 1 client are served.<br>The version parameter shall be composed as follows:<br>version = version_number * [ "_" version_number ]<br>version_number = 1*DIGIT<br>Example: chatbot_version=1_2_3 | N | String |

**Table 48: Query parameters for Directory to Directory query**

The Chatbot Directory of the Chatbot Platform shall return the query result in the JSON format with a HTTP 200 OK response. The schema in Table 49 defines the JSON payload exchanged between the Chatbot Directory of the Chatbot platform and the Service Provider Chatbot Directory. It is possible to get more than one entry with the same Chatbot service ID e.g. different values for some properties (name, email, SMS, tel, website, description) per country that correspond to the same Chatbot service identifier based on Chatbot or Chatbot platform logic.

The description of each JSON object is specified in Table 50.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "botsprofile": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": {
            "type": "string",
            "format": "uri"
```

```
        },
        "name": {
          "type": "string"
        },
        "provider": {
          "type": "string"
        },
        "email": {
          "type": "string"
        },
        "sms": {
          "type": "string"
        },
        "tel": {
          "type": "string"
        },
        "website": {
          "type": "string"
        },
        "icon": {
          "type": "string",
          "format": "uri"
        },
        "iconfingerprint": {
          "type": "string"
        },
        "description": {
          "type": "string"
        },
        "businessLocation": {
          "type": "string"
        },
        "category": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "co": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "cl": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "version": {
          "type": "string"
        },
        "verification-signatures": {
          "description":"RFC7515 (JWS) general JWS JSON serialisation",
          "type": "object"
        }
      },
```

```
          "required": [
            "id",
            "name",
            "version"
          ]
        }
      },
      "itemsReturned": {
        "type": "integer"
      },
      "startIndex": {
        "type": "integer"
      },
      "totalItems": {
        "type": "integer"
      }
    }
    "required": [
      "botsprofile"
    ]
}
```


**Table 49: JSON Schema for Chatbot Directory payloads**

| JSON | Description |
|---|---|
| id | Chatbot Service ID |
| name | Chatbot Service Name |
| provider | Chatbot Provider Name |
| email | Chatbot email address |
| sms | Chatbot SMS address. It can be a short or long code. |
| tel | Chatbot Call-back phone number |
| website | Chatbot website |
| icon | Chatbot Service Icon. Link to the Chatbot icon image |
| iconfingerprint | Fingerprint of the file providing the Chatbot icon image |
| description | Chatbot Service Description |
| businessLocation | Chatbot business location |
| category | Chatbot Category |
| co | Chatbot supported country or network where:<br>• Country representation is using Alpha-2 country code as defined in ISO3166 used when the Chatbot is supported in all the operators of that country<br>• Network representation is using <MCCMNC> combination. Whereby MCC and MNC 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. |
| cl | Chatbot supported language encoded based on [ISO 639-1] definitions. |

| | |
|---|---|
| version | Chatbot version supported by the Chatbot. The version parameter shall be composed as follows:<br>version = version_number * [ "_" version_number ] version_number = 1*DIGIT<br>Example: chatbot_version=1_2_3 |
| verification-signatures | Provides a JSON Web Signature (JWS) in general JWS JSON serialisation format according to [RFC7515] with non-detached payload that can be used to verify the Chatbot Directory entry |
| itemsReturned | The number of query objects returned |
| startIndex | The index of the first query result in the current set of query results |
| totalItems | The number of query objects available |

**Table 50: Description of the Chatbot Directory query result**

#### 3.6.3.2.1 Verification of Chatbot Directory entries

The Service Provider should verify that the entries in the Chatbot Directory query result comply with the information that has been verified by a Chatbot Verification Authority that is recognized by the MNO to perform verifications of Chatbots as follows:

1. If the entry for the Chatbot does not include a *verification-signatures* property, the Chatbot shall be considered as an unverified Chatbot and no further processing to verify whether the Chatbot has been verified by a recognized Chatbot Verification Authority shall be done.
2. If the *verification-signatures* property of the entry for the Chatbot does not correspond to a JWS in the General JWS JSON Serialization defined in section 7.2.1 of [RFC7515], the Chatbot shall be considered as an unverified Chatbot and no further processing to verify whether the Chatbot has been verified by a recognized Chatbot Verification Authority shall be done.
3. If the *verification-signatures* property of the entry for the Chatbot does not include a "*payload*" property (i.e. it is using a detached payload according to Annex F of [RFC7515]), the Chatbot shall be considered as an unverified Chatbot and no further processing to verify whether the Chatbot has been verified by a recognized Chatbot Verification Authority shall be done.
4. If (after Base64url-decoding) the "*payload*" property of the "*verification-signatures*" property of the entry for the Chatbot does not correspond to a JSON object with at least the properties "*id*" and "*name*", the Chatbot shall be considered as an unverified Chatbot and no further processing to verify whether the Chatbot has been verified by a recognized Chatbot Verification Authority shall be done.
5. Otherwise, the Service Provider shall verify the different signature objects in the "*signatures*" property as specified in section 5.2 of [RFC7515] taking into account the following restrictions:

    a) If the "*alg*" header parameter provided in the JWS Protected Header is not set to one of the algorithms defined as "Recommended" or "Recommended+" in section 3.1 of [RFC7518], the digital signature shall not be considered.
    b) If the signature object does not provide in the JWS Protected Header a "*crit*" Header parameter defined in section 4.1.11 of [RFC7515] including the value "*botvfexpires*" among the values, the digital signature shall not be considered.

c) If the signature object does not contain a "*botvfexpires*" property in the JWS Protected Header set to a date in the future, the digital signature shall not be considered.
d) If the signature object does not provide a "*kid*" property in the unprotected header as reference to the key, the digital signature shall not be considered
e) If the value of the "*kid*" property in the unprotected header of the signature object does not correspond to a key of a recognized Verification Authority, the digital signature shall not be considered

NOTE: The process through which the Service Provider is made aware of the "kid" value of the Verification Authorities that it recognizes is out of scope of this specification.

6. If at least one of the considered digital signatures was successfully verified, the Service Provider shall compare the data in the JSON object provided in the "*payload*" property to the data in the corresponding entry for the Chatbot in the Chatbot Directory query result as follows:

    a) If the value of the "*id*" property in the JSON object provided in the "*payload*" property does not match the value of the "*id*" property of the entry in the Chatbot Directory query result, the verification shall be considered to have failed
    b) If the value of the "*name*" property in the "*payload*" property does not match the value of the "*name*" property of the entry in the Chatbot Directory query result, the verification shall be considered to have failed
    c) If the entry in the Chatbot Directory query result had an "*iconfingerprint*" property and the JSON object provided in the "*payload*" property does not have an "*iconfingerprint*" property in the entry in the Chatbot Directory query result, the verification shall be considered to have failed.
    d) If the JSON object provided in the "*payload*" property provides an "*iconfingerprint*" property with a value and the entry in the Chatbot Directory query result does not provide an "*iconfingerprint*" property, the verification shall be considered to have failed.
    e) If the entry in the Chatbot Directory query result has an "*iconfingerprint*" property and the JSON object provided in the "*payload*" property has an "*iconfingerprint*" property, but their values do not match, the verification shall be considered to have failed.
    f) Otherwise, the verification of the "*payload*" property shall be considered to have succeeded

7. If in step 6 the verification of the "*payload*" property was successful, the verification of the entry for the Chatbot in the Chatbot Directory query result shall be considered to be successful if either of the following conditions is fulfilled:

    - No "*icon*" and "*iconfingerprint*" properties were included for the Chatbot in the Chatbot Directory query result, or
    - Both "*icon*" and "*iconfingerprint*" properties were included and the value of the "*iconfingerprint*" property matches a SHA-256 hash of the file referred to by the URL in the "*icon*" property in the Chatbot Directory result whereby the result of

the hash is represented as a hexadecimal string in lowercase without a "0x" prefix.

Otherwise, if those conditions are not fulfilled, the verification of the entry for the Chatbot in the Chatbot Directory query result shall be considered to have failed.

#### 3.6.3.3 Service Provider Client Configuration for Chatbots requiring specific management

A Service Provider may want to indicate that some Chatbots shall require specific management on the client side (e.g. blacklisted as Spam senders, providing critical communication).

If the Service Provider wishes to have specific management for Chatbots, the Service Provider shall provide the clients with a URL from which they can retrieve a list of Chatbots requiring specific management. This URL shall be configured through the SPECIFIC CHATBOTS LIST client configuration parameter defined in section A.1.3. If the URL is not configured, the client shall assume that there are no Chatbots that require specific management.

If the configured URL does not resolve through DNS, the client shall

- assume that there are no Chatbots that require specific management and
- verify the availability of the configured URL after every restart of the device or client.

If the URL is available, the client shall send a HTTP GET request to retrieve the file using the URL from the SPECIFIC CHATBOTS LIST client configuration parameter. If receiving a 200 OK response to the request, the client shall

1. Store the Etag and Cache-Control directive values associated with the resource according to the procedures of [RFC7232];
2. Store the lists of URIs of Chatbots requiring specific management which will be formatted as described in this section;
3. Whenever the validity of the lists of URIs of Chatbots requiring specific management expires (based on the Cache-Control directive received in the last retrieval stored in steps 1, 3 b)i and 3 c)i, send an HTTP GET request to the configured URL including the stored Etag in a if-none-match header according to the procedures of [RFC7232].

    a) If the configured URL no longer resolves through DNS, the client shall

        i. verify availability of the configured URL after every restart of the device or client and
        ii. delete the lists of URIs of Chatbots requiring specific management if the configured URL has failed to resolve through DNS for 10 consecutive attempts.

    b) If receiving a HTTP 200 OK response, the client shall

        i. Store the new Etag and Cache-Control directive values according to the procedures of [RFC7232];
        ii. Store the provided lists of URIs and use them according to the procedures defined for each list.

c) If receiving a HTTP 304 response, the client shall

   i. Store the new Etag and Cache-Control directive values according to the procedures of [RFC7232];
   ii. Continue using the existing lists of URIs and use them according to the procedures defined for each list.

d) If receiving another response, the client shall

   i. Repeat the procedure in step 3 at the next client or device restart
   ii. Continue using the existing lists of URIs and use them according to the procedures defined for each list.

For the Expires header, similarly to section 3.6.4.1.2, the Service Provider shall ensure that the entity that maintains the lists of Chatbots requiring specific management returns responses with Expires header value set to a date in the past. This will ensure that legacy proxies do not attempt to cache the content.

When data is switched off (see section 2.8.1.5), this interface shall remain available if device management is configured as cellular switched data off exempted service via the client configuration parameters Device Management over PS data off exemption or Device Management over PS data off roaming exemption defined in section A.1.14.

If provisioning is not available, the client shall perform step 3 of the above procedure at the earliest opportunity (e.g. when data is switched on again or when connecting over Wi-Fi).

The lists of URIs of Chatbots requiring specific management shall be provided as a document of the Mime-type text/plain providing sub lists of URI entries. Each sub list shall begin with a specific marker:

```
LIST:<name of the list><CRLF>
<CRLF>
```

where <name of the list> is a distinct identifier, identifying the sub list. This identifier is defined in the section describing the specific management that is required.

Each sub list of URIs is constituted of URI entries separated by new lines (i.e. CRLF, resulting in one URI per line).

- The URI entries themselves shall be any of the following:
  - A SIP URI or
    In this case the entry will start with sip: or sips:
  - a tel URI or
    In this case the entry will start with tel:
  - a regular expression matching either of those URI types.
    In this case the entry will start with an exclamation mark character (i.e. ‘!’). The subsequent string shall be interpreted by the client to match a URI string using Portable Operating System Interface (POSIX) extended regular expression (see [POSIX]) e.g. !^(sip:.*-publisher\@botplatform\.example\.com)
- Entries that do not conform to these formats shall be ignored.

A sub list may be empty which shall be handled in the same way as when the sub list was not provided at all.

A client shall ignore any sub list for which the identifier (i.e. <name of the list> ) is unknown.

#### 3.6.3.4 Deep Linking

Deep links are URLs that can take RCS users directly to the Chatbots conversations with specific contents. Deep links can be embedded in a webpage, a mobile application, or a QR code. The usage of deep links is out of the scope of this document.

Because SMS is supported by most mobile devices, the “sms” URI scheme defined in [RFC5724] shall be extended to construct the deep links to initiate a Chatbot conversation from an RCS user. In addition to the “body” query parameter defined in [RFC5724], the following parameters may also be added as part of the deep link:

- an optional “service_id” parameter which is the Chatbot service identifier defined in section 2.5.4.1 without the “sip:” URI scheme.
- an optional “suggestions” parameter which is the base64 URL encoded suggestions JSON object defined in section 3.6.10.3.

If the Chatbot does not have an SMS number, the Chatbot service identifier defined in section 2.5.4.1 shall be used without the "sip:" URI scheme as value for the “sms-recipient” part of the URI that is defined in [RFC5724]. However, in such a case, a non-RCS client may not be able to handle this deep link.

When the deep link is activated, if the RCS client is registered:

- the RCS client shall first discover the contact's RCS capabilities as specified in sections 2.6.1. If a “service_id” parameter is included in the deep link, the SIP URI included in the “service_id” parameter shall be used as target for capability discovery; otherwise, the phone number provided in the “sms-recipient” part of the deep link as defined in [RFC5724] shall be used to construct a tel URI that is used for capability discovery.
- If the deep link includes a “body” or “suggestions” parameter, the RCS client shall prepare a message for the user to send to the contact. If a "suggestions" parameter is available and the contact is RCS capable, the RCS client shall ignore the "body" parameter and show an initial Suggested Chip List based on the JSON object that is obtained after base64 URL decoding the value of the “suggestions” parameter of the deep link; otherwise, the value of the “body” parameter as defined in [RFC5724] shall be used as body for an RCS or SMS message depending on whether the contact is RCS capable.
- If the deep link does not include a “body” or “suggestions” parameter, the RCS client shall prepare an empty message with the contact filled with the value of the “service_id” or “sms-recipient” parameter. In this case, the user is ready to manually compose the message to be sent.

When the deep link is activated, if the RCS client is not registered and if the deep link does include an SMS number in the “sms-recipient” part of the URI that is defined in [RFC5724], the RCS client shall handle the URI as defined in [RFC5724], ignoring the "service_id" and "suggestions" parameters.

Some non-normative examples are given in the following:

- if the Chatbot is assigned an SMS number (+15012011657) and
- the Chatbot service ID is “bot@botplatform.example.com”, and
- the Chatbot wants the user to send a message of “tell me about checking accounts” to start the conversation

  then the deep link URL can be:

  ```
  sms:+15012011657?service_id=bot%40botplatform.example.com&body=
  tell%20me%20about%20checking%20accounts.
  ```

  When this deep link is activated by a RCS client, the message with the Chatbot, identified by the service ID, will be prepared for the user to send via RCS; when this deep link is activated by a non-RCS client, the message with the SMS-message-capable phone number will be prepared for the user to send via SMS.

- if the Chatbot does not have the SMS-message-capable phone number, then the deep link URL can be:

  ```
  sms:bot%40botplatform.example.com?body=tell%20me%20about%20checking%2
  0accounts.
  ```

- if the Chatbot wants the user to start the conversation with a suggested reply

```
"suggestions": [{
   "reply": {
     "displayText": "Open a checking account",
     "postback": {
       "data": "reply=open-checking-account;source=website.com;campaign=chk-act-cmpgn-14"
     }
   }
 }
]
```

  then the deep link URL can be:

  ```
  sms:+15012011657?service_id=bot%40botplatform.example.com&suggestions
  =InN1Z2dlc3Rpb25zIjogW3sKICAgICJyZXBseSI6IHsKICAgICAgImRpc3BsYXlUZXh0
  IjogIk9wZW4gYSBjaGVja2luZyBhY2NvdW50IiwKICAgICAgInBvc3RiYWNrIjogewogI
  CAgICAgICJkYXRhIjogInJlcGx5PW9wZW4tY2hlY2tpbmctYWNjb3VudDtzb3VyY2U9d2
  Vic2l0ZS5jb207Y2FtcGFpZ249Y2hrLWFjdC1jbXBnbi0xNCIKICAgICAgfQogICAgfQo
  gIH0KXQ==
  ```

### 3.6.4 Chatbot Information

This section defines the procedures for the retrieval of Chatbot Information accessible from within the messaging implementation.

#### 3.6.4.1 Chatbot Information Retrieval

##### 3.6.4.1.1 Procedure of constructing botinfo query URL

Chatbot Information is maintained in and provided by the Chatbot Information Function. To retrieve Chatbot Information of a Chatbot, the client shall send a HTTP GET request to the Chatbot Information Function using the botinfo URL formatted as defined in Table 51.

```
https://<root_domain>/bot?<set_of_query_parameters>
```

**Table 51: Composition of the botinfo URL**

The client shall use the URL scheme “https”, i.e. it shall establish a secured connection to the botinfo URL host.

If the configuration parameter BOTINFO FQDN ROOT as defined in section A.1.3 is present in the client configuration, then the value of “root_domain” of the botinfo URL shall be taken from the value of the configuration parameter.

Otherwise, if the configuration parameter is not present, the value of “root_domain” shall take the value of the domain part of the Chatbot service identifier SIP URI defined in section 2.5.4.1.

The client shall use the HTTPS standard port.

The client shall include in the botinfo URL a path with the value “bot”

The client shall include in the botinfo URL the "`set_of_query_parameters`" using the parameters defined in Table 52 using the *application/x-www-form-urlencoded* encoding defined in [HTML-4.0].

| Query Parameter | Description | Mandatory | Format |
|---|---|---|---|
| id | Chatbot service Identifier SIP URI. It shall not contain URL parameters and shall be encoded respecting the definitions of [RFC3986]. | Y | String |
| hl | It indicates the current device language setting. Its format shall follow the [ISO 639-1] definitions. | Y | String |
| ho | Home operator of the device and this shall be represented as <MCCMNC> combination. Whereby MCC and MNC 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. | Y | String |
| v | The version indicates the JSON format version of the Chatbot information data that the client is able to process. It shall be set to 2 for clients complying with this version of this specification. | Y | String |

**Table 52: Query parameters of the botinfo URL**

The client shall connect to the Chatbot Information Function in accordance with the values of the scheme and authority of the botinfo URL and send a HTTP GET request using the botinfo URL resulting from the procedures above.

The client shall specify the botinfo JSON format version that it is able to process. If the version is omitted or v=1, the Chatbot Information Function shall return Chatbot information data in the JSON format that complies with version 8.0 of this document i.e. [PRD-RCC.07v8.0]. If the version is v=2, the Chatbot Information Function shall return Chatbot information data in the JSON format as defined in section 3.6.4.1.3. To provide backward

compatibility, an RCS client shall be able to process Chatbot information data in the JSON format that complies with version 8.0 of this document i.e. [PRD-RCC.07v8.0].

#### 3.6.4.1.2 Procedure of handling botinfo query response

If the client receives in result of processing of the request for the initial Chatbot Information retrieval:

- any HTTP 5XX error response other than HTTP 503 Internal Server error with a Retry-after header or HTTP 4XX error response other than 408:
  - If the request is initiated as the result of a Chatbot attempting to contact a user for the first time:
    - The client shall not attempt any retry request towards the Chatbot Information Function and it shall not apply any UX procedures for Chatbot initiated conversations.
  - If the request is initiated as a result of the user attempting to retrieve more information for a Chatbot that the client has not interacted with before:
    - The client shall not attempt any retry request towards the Chatbot Information Function and it shall notify the user.
- an HTTP 503 Internal Server error response with a Retry-After header:
  - The client shall retry the request towards the Chatbot Information Function. The recommended value to retry shall be specified in the “Retry-After” header. The client shall retry for a maximum of three times.
  - If a request is processed successfully after retry, the procedures described for the HTTP 200 OK response shall apply.
  - If the last retry of the request fails, the client shall consider the transaction failed and the procedures described for any HTTP 5XX error response other than HTTP 503 Internal Server error with a Retry-after header or HTTP 4XX error response other than 408 shall apply.
- an HTTP 408 error response or no response (Chatbot Information retrieval request timeout):
  - The client shall manage the procedures locally on the device. If for the failure of a request a retry is applicable, the client shall retry by sending the same request again. The client shall retry for a maximum of three times.
  - If a request is processed successfully, the procedures described for the HTTP 200 OK response shall apply.
  - If the last retry of the request fails, the client shall consider the transaction failed and the procedures described for any HTTP 5XX error response other than HTTP 503 Internal Server error with a Retry-after header or HTTP 4XX error response other than 408 shall apply
- An HTTP 200 OK response, the client shall

- store the Etag and Cache-Control mandatory header values according to the procedures of [RFC7232] (see Table 53).
- Store the Chatbot Information data, returned by the Chatbot Information Function.
- for a request initiated as the result of a Chatbot attempting to contact a user for first time, apply any UX procedures for Chatbot initiated conversations.

| Header | Value |
|---|---|
| Date | Server current date. |
| Expires | A date in the past to ensure that legacy HTTP/1.0 proxies do not try to cache the content, e.g. Mon, 01 Jan 1990 00:00:00 GMT |
| Cache-Control | private, max-age = <days_in_seconds>. HTTP/1.1 proxies shall not cache the content either due to the private directive. Max-age value is determined by the bilateral contract between Chatbot and Chatbot Platform that is out of the scope of this specification. |
| ETag | <entity_tag_value> assigned by the server. |

**Table 53: Cache control headers for Chatbot Information**

Whenever the Chatbot Information needs to be refreshed, i.e. when it is required (for example, when the user opens the conversation with the Chatbot) and found to be out of date based on the *Cache-Control* directive received during the last retrieval, the client shall send an HTTP request including the stored ETag in a *if-none-match* header according to the procedures of [RFC7232].

If the client receives in result of processing the Chatbot Information refresh request:

- Any HTTP 5XX or HTTP 4XX error response or no response the client shall
  - continue using the existing Chatbot information until the next client trigger and
  - if the Chatbot was considered to be verified (see section 3.6.4.2), remove the verified status of the Chatbot if the Chatbot Information refresh request fails 3 times (i.e. 3 client triggers) in succession.
- an HTTP 304 NOT_MODIFIED response, the client shall:
  - store the Etag and Cache-Control directive value according to the procedures of [RFC7232]
  - continue using the existing Chatbot Information
- an HTTP 200 OK response, the client shall:
  - store the Etag and Cache-Control directive value according to the procedures of [RFC7232]
  - store the provided Chatbot Information data

#### 3.6.4.1.3 Chatbot Information data

The Chatbot Information data is provided in the JSON format. Part of the whole payload is represented using the schema defined in OMA CAB Personal Contact Card (PCC) document [CAB_TS], and an “organization” type PCC shall be used. When saving the Chatbot Information to the Contacts, only the information that is part of the PCC would be

considered. The mapping between the Chatbot Information elements defined in section 15.2 of [PRD-RCC.71] and the PCC is specified in Table 54. The client shall gracefully ignore any JSON object which cannot be recognized.

| Information | PCC Element | Identifier |
|---|---|---|
| Service name | org-name.display-name<br>NOTE: recommended to be a maximum of 100 characters | org-name with org-name-type equal to “OfficialName” |
| Service ID | comm-addr.uri-entry.addr-uri | uri-entry with label equal to “ServiceID” |
| SMS | comm-addr.uri-entry.addr-uri | uri-entry with label equal to “SMS” |
| Call-back phone number | comm-addr.tel.tel-nb.tel-str | tel-type is set to “Work” |
| Service icon | media-list.media-entry.media.media-url | media-entry with label equal to “Icon” |
| Service description | a customized string type element named “org-description” in org-details.<br>NOTE: up to 500 characters. | |
| Category | category-list.category-entry<br>NOTE: recommended to be a maximum of 15 categories in the category-list, with a maximum of 50 characters per category-entry. | |

**Table 54: Mapping between Chatbot Information and PCC**

For the Service icon as indicated in the schema in Table 55 a URL will be provided and a fingerprint could be included that can be used to verify the integrity of the Service icon file after download. The fingerprint shall be a SHA-256 hash of the icon file that the link refers to, provided in lowercase hexadecimal representation without "0x" prefix.

This part with PCC information is complemented with further information on the Chatbot defined in section 15.2 of [PRD-RCC.71]:

- Version Number of the Chatbot Information version to which the provided information conforms. It is provided in the “version” property of the “botinfo” object in the JSON object providing the Chatbot Information. If the "version" property or "botinfo" object is absent, it shall be assumed that the Chatbot Information complies with version 8.0 of this document (i.e. [PRD-RCC.07v8.0]).
- Provider Name provided as a string recommended to be a maximum of 100 characters in the "provider" property of the "botinfo" object in the JSON object providing the Chatbot Information
- An email-address provided in the "email" property of the "botinfo" object in the JSON object providing the Chatbot Information
- A website provided in the "website" property of the "botinfo" object in the JSON object providing the Chatbot Information

- A link to a Terms and Conditions page provided in the "TCPage" property of the "botinfo" object in the JSON object providing the Chatbot Information
- A colour related to the Chatbot provided as a base64 url encoded hexadecimal RGB colour representation in the "colour" property of the "botinfo" object in the JSON object providing the Chatbot Information
- A link to an image recommended to be a maximum of 200 KB in jpg, png or GIF format that can be used as background provided as a URL in the "backgroundImage" property of the "botinfo" object in the JSON object providing the Chatbot Information
- A Chatbot business location provided as a string recommended to be a maximum of 200 characters in the "address" property of the "botinfo" object in the JSON object providing the Chatbot Information
- Verification information (see also section 3.6.4.2) which can take one of the following two formats:

  - An object confirming whether the Chatbot was verified and providing additional metadata on the verification
    This shall be the only format used on the UNI.
  - An object providing signatures in JWS format that can be used to verify the information as specified in section 3.6.4.2.2
    This format shall be the only format used between Chatbot Platform and Service Provider network. Information in this format shall be ignored when provided on the UNI.

NOTE: If Verification information is provided in this format, the verification procedures defined in sections 3.6.4.2.1 and 3.6.4.2.2 mean that a fingerprint must be provided to ensure a successful validation of this signature if an icon is provided.

The JSON schema for Chatbot Information is defined in Table 55.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",

  "definitions": {
    "tel-nb": {
       "type": "object",
       "properties": {
          "tel-str": {
             "type": "string"
          }
       },
       "required": ["tel-str"]
    },
    "tel": {
       "type": "object",
       "properties": {
          "label": {
            "type": "string"
          },
          "tel-nb": {
            "$ref": "#/definitions/tel-nb"
          },
```

```
          "tel-type": {
            "type": "string"
          }
        },
        "required": ["label", "tel-nb", "tel-type"]
      },
      "uri-entry": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "addr-uri": {
              "type": "string",
              "format": "uri"
            },
            "addr-uri-type": {
              "type": "string",
              "enum": ["SIP-URI", "Other"]
            },
            "label": {
              "type": "string",
              "enum": ["ServiceID", "SMS"]
            }
          },
          "required": ["addr-uri", "addr-uri-type", "label"]
        }
      },
      "comm-addr": {
        "type": "object",
        "properties": {
          "tel": {
            "$ref": "#/definitions/tel"
          },
          "uri-entry": {
            "$ref": "#/definitions/uri-entry"
          }
        },
        "required": ["tel", "uri-entry"]
      },
      "media": {
        "type": "object",
        "properties": {
          "media-url": {
            "type": "string",
            "format": "uri"
          },
          "fingerprint": {
            "type": "string"
          }
        },
        "required": ["media-url"]
      },
      "media-entry": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "label": {
```

```
          "type": "string",
          "enum": ["Icon"]
        },
        "media": {
          "$ref": "#/definitions/media"
        },
        "media-content": {
          "type": "string",
          "enum": ["Logo", "Other"]
        }
      },
      "required": ["label", "media", "media-content"]
    }
  },
  "org-name": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "display-name": {
          "type": "string"
        },
        "org-name-type": {
          "type": "string",
          "enum": ["OfficialName"]
        }
      },
      "required": ["display-name", "org-name-type"]
    }
  },
  "address-entry": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "addr-string": {
          "type": "string"
        },
        "label": {
          "type": "string"
        }
      },
      "required": ["addr-string", "label"]
    }
  },
  "category-entry": {
    "type": "array",
    "items": {
      "type": "string"
    }
  },
  "verification-info": {
    "type": "object",
    "oneOf": [{
      "properties" : {
        "verified": {
          "type": "boolean"
        },
```

```
        "verified-by": {
          "type": "string"
        },
        "expires": {
          "type": "string",
          "format": "date-time"
        }
      },
      "required": ["verified", "verified-by", "expires"]
    },
    {
      "properties": {
       "verification-signatures": {
         "description": "RFC7515 (JWS) general JWS JSON serialisation",
         "type": "object"
       }
      }
    }
    ]
  }
},

"type": "object",

"properties": {
  "botinfo": {
    "type":"object",
    "properties": {
      "pcc": {
        "type": "object",
        "properties": {
          "org-details": {
            "type": "object",
            "properties": {
              "comm-addr": {
                "$ref": "#/definitions/comm-addr"
              },
              "media-list": {
                "type": "object",
                "properties": {
                  "media-entry": {
                    "$ref": "#/definitions/media-entry"
                  }
                }
              },
              "org-name": {
                "$ref": "#/definitions/org-name"
              },
              "org-description": {
                "type": "string",
                "maxLength": 500
              },
              "category-list": {
                "type": "object",
                "properties": {
                  "category-entry": {
                    "$ref": "#/definitions/category-entry"
                  }
```

```
                }
              }
            }
          },
          "pcc-type": {
            "type": "string"
          }
        }
      },
      "version":{
        "type":"string"
      },
      "provider":{
        "type":"string"
      },
      "email":{
        "type":"string",
        "format":"email"
      },
      "colour":{
        "type":"string",
        "description": "base64 url encoded colour representation"
      },
      "backgroundImage":{
        "type":"string",
        "format":"uri"
      },
      "website":{
        "type":"string",
        "format":"uri"
      },
      "TCPage":{
        "type":"string",
        "format":"uri"
      },
      "address": {
        "type": "object",
        "properties": {
          "address-entry": {
            "$ref": "#/definitions/address-entry"
          }
        }
      }
    },
    "required": ["pcc"]
  }
 },
 "bot-verification": {
   "$ref": "#/definitions/verification-info"
 },
 "required": ["botinfo"]
}
```

**Table 55: JSON schema for Chatbot Information**

A non-normative example Chatbot Information request and response between Chatbot Platform and Service Provider Network can be found in Table 56

```
GET /bot?id=foo.bar%40botplatform.examplebot.com&&hl=en&ho=370270&v=2
HTTP/1.1
Host: botinfo.botplatform.com
Accept: application/json
If-None-Match: 1

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Mon, 17 Apr 2017 20:00:00 GMT
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Cache-Control: private, max-age=86400
ETag: 2

{
  "botinfo":{
    "pcc": {
      "pcc-type": "organizaton",
      "org-details": {
        "org-name": [{
          "display-name": "Service Name",
          "org-name-type": "OfficialName"
        }
        ],
        "comm-addr": {
          "uri-entry": [{
            "addr-uri-type": "SIP-URI",
            "addr-uri": "sip:serviceid@example.com",
            "label": "ServiceID"
          },
          {
            "addr-uri-type": "Other",
            "addr-uri": "sms:+15105550101",
            "label": "SMS"
          }],
          "tel": {
            "tel-type": "Work",
            "tel-nb": {
              "tel-str": "1-800-555-1213"
            },
            "label": "Callback"
          }
        },
        "media-list": {
          "media-entry": [{
            "media-content": "Logo",
            "media": {
              "media-url": "http://example.com/myLogo.png",
              "fingerprint": "73a982e7930460d3c047c2429f35cdeeebba6e86eed259d50c40ef27b8a87711"
            },
            "label": "Icon"
          }]
        },
        "category-list": {
          "category-entry": [
            "News", "Finance"
          ]
```

```
      },
      "web-resources": {
        "web-entry": [{
          "url": "http://example.com/index.html",
          "label": "Website"
        },
        {
          "url": "http://example.com/tc.html",
          "label": "TCPage"
        }]
      },
      "org-description": "Example service description"
    }
  },
  "version": "2",
  "provider": "Brand X",
  "colour": "IzAwZmYwMA==",
  "backgroundImage":"http://example.com/myBackground.png",
  "website":"http://example.com/index.html",
  "TCPage": "http://example.com/tc.html",
  "address": {
    "address-entry": [{
      "addr-string": "XYZ Corporation,111 Park Avenue,Huntsville AL 11111, USA",
      "label": "BusinessLocation"
    }]
  },
 },
 "bot-verification": {
  "verification-info": {
    "verification-signatures": {
      "payload":
"eyJpY29uZmluZ2VycHJpbnQiOiI3M2E5ODJlNzkzMDQ2MGQzYzA0N2MyNDI5ZjM1Y2RlZWViYmE2ZTg2ZWVkMjU5ZDUwYzQwZWYyN2I4YTg3NzExIiwiaWQiOiJzaXA6c2VydmljZWlkQGV4YW1wbGUuY29tIiwibmFtZSI6IlNlcnZpY2UgTmFtZSJ9",
      "signatures": [{
        "protected":
"eyJhbGciOiJFUzI1NiIsImNyaXQiOlsiYm90dmZleHBpcmVzIl0sImJvdHZmZXhwaXJlcyI6IjIwMTgtMDItMjJUMTQ6MjI6NTJaIn0=",
        "header":{
          "kid":"ec45e387-b742-436a-7312-e37e9a41148d"
        },
        "signature":
          "DpNiU3ljbEg87O8PLifUAqOyKAM6-Xx-E7GawxeppmNFCgfTiDoi7djxLa8ILOSApmWNgfKTUJqPP3-Kg6NY2R"
      },
      {
        "protected":
"eyJhbGciOiJFUzI1NiIsImNyaXQiOlsiYm90dmZleHBpcmVzIl0sImJvdHZmZXhwaXJlcyI6IjIwMTgtMDQtMzBUMDQ6NTM6MTVaIn0=",
        "header": {
          "kid":"ec4c03a7-ca54-4719-8327-b1c4e874db0d"
        },
        "signature":
          "DiNiUpilbEg8e88OLifUZqOyKsD6-Xx-E7GaIUenfcNFCgfTiRop7dj8La78LOSApmSNmgKTUJUPP3-Kp6RS2t"
      }
      ]
    }
```

```
   }
  }
}
```

**Table 56: Example Chatbot Information request and response between Chatbot Platform and Service Provider Network**

A non-normative example Chatbot Information request and response between Service Provider Network and Client can be found in Table 57

```
GET /bot?id=foo.bar%40botplatform.examplebot.com&&hl=en&ho=370270&v=2
HTTP/1.1
Host: botinfo.operator.com
Accept: application/json
If-None-Match: 1

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Mon, 17 Apr 2017 20:00:00 GMT
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Cache-Control: private, max-age=86400
ETag: 2

{
  "botinfo":{
    "pcc": {
      "pcc-type": "organizaton",
      "org-details": {
        "org-name": [{
          "display-name": "Service Name",
          "org-name-type": "OfficialName"
         }
        ],
        "comm-addr": {
          "uri-entry": [{
            "addr-uri-type": "SIP-URI",
            "addr-uri": "sip:serviceid@example.com",
            "label": "ServiceID"
          },
          {
            "addr-uri-type": "Other",
            "addr-uri": "sms:+15105550101",
            "label": "SMS"
          }],
          "tel": {
            "tel-type": "Work",
            "tel-nb": {
              "tel-str": "1-800-555-1213"
            },
            "label": "Callback"
          }
        },
        "media-list": {
          "media-entry": [{
            "media-content": "Logo",
            "media": {
```

```
            "media-url": "http://example.com/myLogo.png",
            "fingerprint": "73a982e7930460d3c047c2429f35cdeeebba6e86eed259d50c40ef27b8a87711"
          },
          "label": "Icon"
        }]
      },
      "category-list": {
        "category-entry": [
          "News", "Finance"
        ]
      },
      "web-resources": {
        "web-entry": [{
          "url": "http://example.com/index.html",
          "label": "Website"
        },
        {
          "url": "http://example.com/tc.html",
          "label": "TCPage"
        }]
      },
      "org-description": "Example service description"
    }
  },
  "version": "2",
  "provider": "Brand X",
  "colour": "IzAwZmYwMA==",
  "backgroundImage":"http://example.com/myBackground.png",
  "website":"http://example.com/index.html",
  "TCPage": "http://example.com/tc.html",
  "address": {
    "address-entry": [{
      "addr-string": "XYZ Corporation,111 Park Avenue,Huntsville AL 11111, USA",
      "label": "BusinessLocation"
    }]
  },
 },
 "bot-verification": {
  "verification-info": {
    "verified":true,
    "verified-by":"MNO Accepted Verification Authority",
    "expires":"2020-04-04T23:59:00Z"
  }
 }
}
```


**Table 57: Example Chatbot Information request and response between Service Provider Network and client**

### 3.6.4.2 Verification of Chatbot Information

#### 3.6.4.2.1 Verification Authority Procedures

When a verification authority verifies a Chatbot and considers it as trustworthy, it shall generate a signature using its private key over a JSON object complying with the following schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "id": {
      "type": "string",
      "format": "uri"
    },
    "name": {
      "type": "string"
    },
    "iconfingerprint": {
      "type": "string"
    },
  },
  "required": [
    "id", "name"
  ]
}
```

**Table 58: JSON Schema for Chatbot Verification payloads**

In this JSON object, the different properties shall be set as follows:

| JSON | Description |
|---|---|
| id | Chatbot service ID of the verified Chatbot without URI parameters |
| name | Chatbot name of the verified Chatbot |
| iconfingerprint | Fingerprint of the file providing the Chatbot icon image. The fingerprint shall be a SHA-256 hash of the icon file that was verified, provided in lowercase hexadecimal representation without "0x" prefix. If no icon is available, the iconfingerprint property shall not be provided. |

**Table 59: Mapping of Chatbot data to JSON Schema for Chatbot verification payloads**

This JSON shall be used as payload for a JWS signature. That JWS shall be created as follows:

- Unless the actual formatting of the JSON object used for a particular Chatbot is fixed by a central entity (e.g. Chatbot Platform or Chatbot itself) guaranteeing that object is provided in the exact same format to all verification authorities verifying the Chatbot, the verification authority shall use the following rules for the formatting of the object in the payload:
  - no whitespace or line breaks shall be provided before or after any syntactic elements and
  - the included members shall be ordered lexicographically by the Unicode code points of the member names.

  Example:

```
{"iconfingerprint":"73a982e7930460d3c047c2429f35cdeeebba6e86eed259d50c40ef27b8a87711","id":"sip:serviceid@example.com","name":"Service Name"}
```

**Table 60: Example of Chatbot verification payloads**

The JSON object representation shall then be base64url encoded as defined in [RFC7515] to be used as payload in the JWS

- The Verification Authorities public key shall be identified through the key's fingerprint which will be determined as specified in [RFC7638]. It shall be provided in the JWS in a *kid* property in the unprotected header.
- the JWS Protected header shall include
  - a *crit* header as defined in [RFC7515] with including the value "*botvfexpires*" and
  - a *botvfexpires* property set to the chosen expiry date of the signature, encoded as a JSON string of the date-time format.
- The JWS shall be generated as specified in section 5.1 of [RFC7515] using one of the algorithms specified as "Recommended+" or "Recommended" in section 3.1 of [RFC7518].

#### 3.6.4.2.2 Chatbot Platform Procedure

The Chatbot Platform shall provide the signatures that are generated by the Verification Authorities following the process defined in section 3.6.4.2.1 together with the Chatbot Information as specified in section 3.6.4.1.3.

#### 3.6.4.2.3 Service Provider procedures

To perform the verification of Chatbot Information, the Service Provider shall ensure that it is included in the retrieval path of such information by configuring the clients with a BOTINFO FQDN ROOT configuration parameter defined in section A.1.3 that routes the request to where the verification shall be performed as defined in section 3.6.4.1.1.

The procedure through which a Service Provider verifies Chatbot Information should rely on the procedure defined in section 3.6.4.2.3.1 or on the verification of the Chatbot that was done when receiving the information from the Chatbot Directory defined in section 3.6.3.2.1. To verify the signatures the Service Provider shall support the algorithms defined as "Recommended" and "Recommended+" in section 3.1 of [RFC7518]. The verification of the Chatbot Information based on a recognized verification authority’s signature is recommended. Also, to ensure consistency in the verification status of a given Chatbot across Chatbot Directory query and Chatbot Information retrieval, the same type of verification procedure, i.e., signature-based or otherwise, should be followed in both scenarios. The Service Provider shall

- Construct the Chatbot Information that will be provided to the client by
  - Including the "*botinfo*" property from the Chatbot Information that was received from the Chatbot Platform and
  - when the Chatbot Information has been verified
    - add a *bot-verification* property with a *verification-info* property according to the schema defined in section 3.6.4.1.3
    - add in the *verification-info* property according to the schema defined in section 3.6.4.1.3 a *verified* property set to true and also *verified-by* and *expires* properties providing information on the verification for display purposes.

#### 3.6.4.2.3.1 Signature Verification Procedure

The procedure to verify whether the Chatbot Information provided by the Chatbot Platform matches the information that has been verified by a Chatbot Verification Authority that is recognized by the Service Provider to perform verifications of Chatbots shall be as follows:

1. If the Chatbot Information does not include a *bot-verification* property that has a *verification-info* property with a *verification-signatures* property, the Chatbot Information shall be considered as an unverified and no further processing to verify whether the Chatbot Information has been verified by a recognized Chatbot Verification Authority shall be done.
2. If the *verification-signatures* property does not correspond to a JWS in the General JWS JSON Serialization defined in section 7.2.1 of [RFC7515], the Chatbot Information shall be considered as an unverified and no further processing to verify whether the Chatbot Information has been verified by a recognized Chatbot Verification Authority shall be done.
3. If the *verification-signatures* property does not include a *payload* property (i.e. it is using a detached payload according to Annex F of [RFC7515]), the Chatbot Information shall be considered as an unverified and no further processing to verify whether the Chatbot Information has been verified by a recognized Chatbot Verification Authority shall be done.
4. If (after Base64url-decoding) the *payload* property of the *verification-signatures* property does not correspond to a JSON object with at least the properties *id* and *name*, the Chatbot Information shall be considered as an unverified and no further processing to verify whether the Chatbot Information has been verified by a recognized Chatbot Verification Authority shall be done.
5. Otherwise, the different signature objects in the *signatures* property shall be verified as specified in section 5.2 of [RFC7515] taking into account the following restrictions:

    a) If the "*alg*" header parameter provided in the JWS Protected Header is not set to one of the algorithms defined as "Recommended" or "Recommended+" in section 3.1 of [RFC7518], the digital signature shall not be considered.
    b) If the signature object does not provide in the JWS Protected Header a "*crit*" Header parameter defined in section 4.1.11 of [RFC7515] including the value "*botvfexpires*" among the values, the digital signature shall not be considered.
    c) If the signature object does not contain a *botvfexpires* property in the JWS Protected Header set to a date in the future, the digital signature shall not be considered.
    d) If the signature object does not provide a *kid* property in the unprotected header as reference to the key, the digital signature shall not be considered
    e) If the value of the *kid* property in the unprotected header of the signature object does not correspond to a key of a Verification Authority recognized by the service provider, the digital signature shall not be considered

6. If at least one of the considered digital signatures was successfully verified, the data in the JSON object provided in the *payload* property shall be compared to the data in the corresponding entry for the Chatbot in the Chatbot Information as follows:

    a) If the value of the *id* property in the JSON object provided in the *payload* property does not match the value of the *id* query parameter used to retrieve

the Chatbot Information (see Table 52), the verification shall be considered to have failed

b) If the value of the *name* property in the *payload* property does not match the Service name information in the Chatbot Information (mapped as defined in Table 54), the verification shall be considered to have failed
c) If the Chatbot Information has a *fingerprint* property in the *media* property carrying the Service icon information (mapped as defined in Table 54) and the JSON object provided in the *payload* property does not have an *iconfingerprint* property, the verification shall be considered to have failed.
d) If the JSON object provided in the *payload* property provides an *iconfingerprint* property with a value and the Chatbot Information does not have a *fingerprint* property in the *media* property carrying the Service icon information (mapped as defined in Table 54), the verification shall be considered to have failed.
e) If the JSON object provided in the *payload* property provides an *iconfingerprint* property and the Chatbot Information has a *fingerprint* property in the *media* property carrying the Service icon information (mapped as defined in Table 54), but their values do not match, the verification shall be considered to have failed.
f) Otherwise, the verification of the "*payload*" property shall have succeeded.

7. If in step 6 the verification of the "*payload*" property was successful, it shall be verified whether a property was provided for an icon in the Chatbot Information,

    a) If not, the verification of the Chatbot Information shall be considered to have succeeded
    b) Otherwise, the verification of the Chatbot Information shall be considered to have succeeded when the value of the "fingerprint" property matches a SHA-256 hash of the file referred to by the URL of the icon in the Chatbot Information whereby the result of the hash is represented as a hexadecimal string in lowercase without a "0x" prefix.

#### 3.6.4.2.4 Client Procedures

The client shall consider the Chatbot Information as verified if

- the *verification-info* property of the *bot-verification* property in the Chatbot Information contained a *verified* property set to *true* and
- the BOTINFO FQDN ROOT client configuration parameter has been provided.

In this case, the client shall use the information provided in the *verified-by* and *expires* properties when displaying information to the user about the verification for respectively an indication of the party that has performed the verification and its validity.

The client shall ignore any *verification-signatures* property in the *verification-info* property of the *bot-verification* property in the Chatbot Information.

### 3.6.5 Privacy Protection

#### 3.6.5.1 Anonymization

##### 3.6.5.1.1 Overview

The procedures for Anonymization defined in this section enable an RCS client to communicate with Chatbots while hiding the identity of the user.

It is a Service Provider's decision to enable the Anonymization feature on its clients via the PRIVACY DISABLE client configuration parameter defined in annex A.1.3.

The Anonymization Function (AF) is the functional element in charge of hiding the identity of the user .

The AF is responsible for the association between a user's URI, a Chatbot URI and a token. The AF manages the repository where those associations are stored. How long the token is valid is up to the Service Provider's policy and the user decision on when to reset the token.

The location of the AF is a deployment decision. The AF can be deployed

- in the Service Provider's network (i.e. MNO AF) or
- as part of the Chatbot Platform (i.e. Chatbot Platform AF).

For Chatbot Service traffic, it is up to the client to request privacy (i.e. Anonymization) or not (see section 3.6.5.1.2.1) when contacting a Chatbot. When privacy is requested, an AF shall be triggered.

NOTE: A Service Provider's policy may as well decide to trigger an AF for each SIP request and corresponding SIP response exchanged between a client and a Chatbot Platform.

For Capability exchanges between a client and a Chatbot, it is up to the Service Provider's policy to decide if the AF shall be triggered. If yes, all requests may be processed as requesting Anonymization.

A client can request to delete a token previously used to communicate with a Chatbot via an extension of the CPM system message. To indicate this, the following IARI is defined and shall be used (see section 3.6.5.1.2.3):

*+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcsmaap.tkdel"*

##### 3.6.5.1.2 Client procedures

###### 3.6.5.1.2.1 Requesting Privacy for a Chatbot conversation

A client can request privacy for a Chatbot conversation (see step 5 in section 3.6.8.3).

###### 3.6.5.1.2.2 Receiving a SIP request from a Chatbot

A client can know if privacy for a Chatbot conversation is enabled or not (see section 3.6.8.4).

#### 3.6.5.1.2.3 Deleting a token

Before deleting a token, the client shall ensure that it has sent all disposition notifications associated with the received messages of the conversation using the token.

When a token associated with a Chatbot needs to be deleted, the client shall send a SIP MESSAGE request according to the rules and procedures of [RCS-CPM-CONVFUNC-ENDORS] with the clarifications listed here. In this SIP MESSAGE request, the client:

1. shall set the Request-URI and To header fields to the address of the Chatbot;
2. shall include the address of the originating RCS client that has been authenticated as per [RCS-CPM-CONVFUNC-ENDORS];
3. shall set the P-Preferred-Service header field with the value set to ‘urn:urn-7:3gpp-service.ims.icsi.oma.cpm.systemmsg’;
4. shall include an Accept-Contact header field with the CPM Feature Tag ‘urn:urn-7:3gpp-service.ims.icsi.oma.cpm.systemmsg’;
5. shall add another Accept-Contact header field carrying the IARI tag defined in section 3.6.5.1.1;
6. shall include a User-Agent header field as specified in Annex C.4.1;

The client shall consider the token as deleted only after receiving the corresponding SIP 200 OK to the SIP MESSAGE request as sent above.

When client receives 200 OK to the SIP MESSAGE request and there is an active Chatbot session between client and the Chatbot that is anonymised (i.e. that included the 'tk=on' parameter in the P-Asserted-Identity header of the received SIP INVITE request or SIP 200 OK response), then the client shall close the session by sending in the corresponding SIP dialog(s) a SIP BYE request 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").

Upon receipt of a SIP 500 Server Internal Error response, the client shall consider that the token is not deleted and the client may retry to delete the token.

#### 3.6.5.1.2.4 Linking a token with the User’s MSISDN

A client requesting to link or share the MSISDN of the user with a Chatbot shall establish a Chatbot session with the Chatbot, by sending a SIP INVITE following the procedures of section 3.6.8.3.

The client shall consider that the MSISDN is linked only after receiving the corresponding SIP 200 OK to the SIP INVITE as sent above.

NOTE: If the SIP URI of the user is not based on the user’s MSISDN then the MSISDN will not be shared, but the real user’s identity is shared with the Chatbot.

### 3.6.5.1.3 MNO AF procedures

#### 3.6.5.1.3.1 Processing SIP requests from clients and SIP responses to clients

Upon receiving a SIP request carrying the Chatbot IARI feature tag defined in section 3.6.2.1 included in the Accept-Contact header field and without the Chatbot role as defined in section 3.6.2.3 included in the Contact header, the MNO AF:

1. if the request is a SIP INVITE or a SIP MESSAGE, shall check if a Privacy header with a value “tk” or “tklink” is part of the received request.

    a) If not , the MNO AF shall forward the request unaltered to the Chatbot platform and not continue with the following steps;
    b) Otherwise, continue with the following steps:

2. shall retrieve the URIs of the user and the Chatbot respectively from the P-Asserted-Identity and Request-URI headers;
3. shall check if a token is already associated to the (user URI, Chatbot URI) pair in its repository;

    a) If for any reason the AF cannot access its repository, the MNO AF shall respond to the SIP request with a SIP 500 Server Internal Error response with optionally a Retry-After header set to a locally configured value and shall not proceed with the rest of the steps.
    b) If no token is associated to the (user URI, Chatbot URI) pair,

        i. if a Privacy header with a value “tklink” is part of the received request, the MNO AF shall respond to the SIP INVITE with a SIP 403 Forbidden response with a Warning header with the warning text set to “204 Token Not Found” and the MNO AF shall not proceed with the rest of the steps.
        ii. Otherwise, the MNO AF shall create a token as per section 2.5.4.2 and associate it with the (user URI, Chatbot URI) pair; If for any reason, the MNO AF cannot create a token, the MNO AF shall respond to the SIP request with a SIP 500 Server Internal Error response with optionally a Retry-After header set to a locally configured value and shall not proceed with the rest of the steps;

    c) Otherwise the MNO AF shall use the token associated with the pair;

4. shall act as a SIP B2BUA with the following precisions. The MNO AF:

    a) if the request is a SIP INVITE or a SIP MESSAGE with a Privacy header set to “tk”

        i. shall remove the “tk” value from the Privacy header. If there is no other value in the Privacy header, the AF shall remove the Privacy header.
        ii. shall replace the URIs representing the Public User Identity in the From; P-Asserted-Identity and in any other SIP headers carrying the Public User Identity with a URI formatted as defined in section 2.5.4.2 using the token and the domain of the MNO AF, leaving all URI parameters untouched, in the received SIP request.

    b) if the request is a SIP INVITE with a Privacy header set to “tklink”

        i. shall remove the “tklink” value from the Privacy header. If there is no other value in the Privacy header in the SIP INVITE, the AF shall remove the Privacy header.

ii. shall set in the SIP INVITE another P-Asserted-Identity header field to a URI formatted as defined in section 2.5.4.2 using the token and the domain of the MNO AF.

c) if the request is related to Capability Discovery, i.e. a SIP OPTIONS or SIP SUBSCRIBE request, may according to Service Provider's policy replace the URIs representing the Public User Identity in the From, P-Asserted-Identity and in any other SIP headers carrying the Public User Identity with a URI formatted as defined in section 2.5.4.2 using the token and the domain of the MNO AF, leaving all URI parameters untouched, in the received SIP request.
d) shall send the SIP request.
e) if the request is a SIP INVITE or a SIP MESSAGE with a Privacy header set to "tk", shall add an 'tk' URI parameter (as defined in section 2.5.4.3) set to 'on' to each of the URIs in the P-Asserted-Identity of the related SIP response(s) sent back on the originating leg;
f) Otherwise (i.e. the Privacy header is set to "tklink"), shall add a 'tk' URI parameter (as defined in section 2.5.4.3) set to 'off' to each of the URIs in the P-Asserted-Identity of the related SIP response(s) sent back on the originating leg;

Upon receiving a SIP 200 OK to a SIP INVITE with a Privacy header set to "tklink", the MNO AF:

1. shall invalidate the corresponding token in the repository.
2. If there is an established 1-to-1 Chatbot session between the user and the Chatbot where the MNO AF acts as a B2B UA, the MNO AF shall close the session by sending in the corresponding SIP dialog(s) a SIP BYE request 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. shall forward the SIP 200 OK to the requesting client with "tk" parameter set to "off" in the P-Asserted-Identity header.

Upon receiving a SIP request that is part of a SIP B2BUA dialog (e.g. SIP ACK, SIP BYE, SIP NOTIFY), or a SIP CANCEL, the MNO AF shall:

1. replace the URIs representing the Public User Identity in the From, P-Asserted-Identity (if present) and in any other SIP headers carrying the Public User Identity with a URI formatted as defined in section 2.5.4.2 using the token and the domain of the MNO AF, leaving all URI parameters untouched, in the received SIP request;
2. send the SIP request;
3. for all SIP provisional or final responses to be sent back to the client on the originating leg, if the P-Asserted-Identity header is present, the MNO AF shall add a 'tk' URI parameter (as defined in section 2.5.4.3) set to 'on', to the URI in the P-Asserted-Identity if not already present;

If the SIP transaction is not anonymised, for all SIP provisional or final responses to be sent back to the client on the originating leg, the MNO AF shall add a 'tk' URI parameter (as defined in section 2.5.4.3) set to 'off', to each of the URIs in the P-Asserted-Identity if not already present.

#### 3.6.5.1.3.2 Processing SIP requests from Chatbots and SIP responses to Chatbots

Upon receiving a SIP request carrying the Chatbot IARI feature tag as defined in section 3.6.2.1 included in the Accept-Contact header field and the Chatbot role as defined in section 3.6.2.3 included in the Contact header, the MNO AF shall act as a SIP B2BUA with the following precisions: the MNO AF

1. shall retrieve the URIs of the user and the Chatbot respectively from the Request-URI and P-Asserted-Identity headers;
2. if the URI of the user is in the format defined in section 2.5.4.2, shall check in its repository if the token that is used in the URI is associated to the Chatbot;

    a) If there is no association, the MNO AF shall return a SIP 404 Not Found error response to the Chatbot Platform;
    b) Otherwise, if there is an association, the AF shall use its repository information associated with the token and the Chatbot URI to retrieve the user's URI. The MNO AF shall forward the SIP request, replacing in all relevant SIP headers the URI of the user in the format defined in section 2.5.4.2 with the user's URI and adding a 'tk' URI parameter set to 'on' to the Chatbot URI(s) provided in the P-Asserted-Identity header if not already present. The MNO AF shall use the URI received in the Request URI as the user's identifier used in the SIP response returned on the originating leg.

3. if the URI of the user is not in the format defined in section 2.5.4.2, then the MNO AF shall

    a) add a 'tk' URI parameter set to 'off' to the Chatbot URI(s) contained in the P-Asserted-Identity header if a 'tk' URI parameter is not already present;
    b) send the SIP request

Upon receiving a SIP request that is part of a SIP dialog (e.g. SIP ACK, SIP BYE, SIP NOTIFY), or a SIP CANCEL, the MNO AF shall

1. use its repository information associated with the token and the Chatbot URI to retrieve the user's URI. The MNO AF shall forward the SIP request, replacing in all relevant SIP headers the URI of the user in the format defined in section 2.5.4.2 with the user's URI and adding a 'tk' URI parameter set to 'on' to the Chatbot URI(s) provided in the P-Asserted-Identity header if not already present;
2. send the SIP request.
3. remove the P-Asserted-Identity header in the response towards the Chatbot.

#### 3.6.5.1.3.3 Processing a token deletion

Upon receiving a SIP MESSAGE as defined in section 3.6.5.1.2.3, the MNO AF:

1. shall retrieve the URIs of the user and the Chatbot respectively from the P-Asserted-Identity and Request-URI headers;
2. shall check in its repository if a token is associated with the retrieved user and Chatbot URIs;

a) If there is no association, the MNO AF shall return a SIP 200 OK to the client to the received SIP MESSAGE request and shall not proceed with the rest of the steps
b) otherwise, if there is such association, the MNO AF shall invalidate the token in the repository and shall return a SIP 200 OK to the client to the received SIP MESSAGE request.
c) if for any reason, the MNO AF cannot access its repository, the MNO AF shall respond to the SIP MESSAGE with a SIP 500 Server Internal Error response with optionally a Retry-After header set to a locally configured value.

3. If there is an established 1-to-1 Chatbot session between the user and the Chatbot where the MNO AF acts as a B2B UA, the MNO AF shall close the session by sending in the corresponding SIP dialog(s) a SIP BYE request 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").

#### 3.6.5.1.4 Chatbot Platform AF procedures

##### 3.6.5.1.4.1 Processing SIP requests from clients

Upon receiving a SIP request carrying the Chatbot IARI feature tag defined in section 3.6.2.1 included in the Accept-Contact header field and without the Chatbot role as defined in section 3.6.2.3 included in the Contact header, the AF:

1. shall retrieve the URIs of the user and the Chatbot respectively from the P-Asserted-Identity and Request-URI headers;
2. if the request is a SIP INVITE or a SIP MESSAGE with a Privacy header set to “tk”,

    a) if for any reason (e.g. the AF cannot access its repository, or a token cannot be generated) anonymity cannot be performed, the Chatbot Platform AF shall respond to the SIP request with a SIP 500 Server Internal Error response with optionally a Retry-After header set to a locally configured value and shall not proceed with the rest of the steps.
    b) the Chatbot Platform AF shall check if the (user URI, Chatbot URI) pair is known in its repository;

        i. If the pair is not known, the AF shall create a token and associate it with the (user URI, Chatbot URI) pair;
        ii. otherwise, the AF shall use the token associated with the pair.

    c) shall provide the token to the Chatbot as the user’s identity.
    d) shall send a SIP response according to section 3.6.8.2 with the following clarification:

        i. a ‘tk’ URI parameter (as defined in section 2.5.4.3) set to ‘on’ shall be added to each of the URIs in the P-Asserted-Identity header field.

3. if the request is a SIP INVITE with a Privacy header set to “tklink”,

    a) if for any reason the AF cannot access its repository, the Chatbot Platform AF shall respond to the SIP request with a SIP 500 Server Internal Error

response with optionally a Retry-After header set to a locally configured value and shall not proceed with the rest of the steps.

b) shall check if the (user URI, Chatbot URI) pair is known in its repository;

   i. If the pair is not known, the Chatbot platform AF shall respond to the SIP INVITE with a SIP 403 Forbidden with a Warning header with the warning text set to “204 Token Not Found” and shall not proceed with the rest of the steps;
   ii. otherwise, the Chatbot Platform shall
      - indicate to the Chatbot that the user’s URI and the token associated to the pair represent the same user.
        How the Chatbot platform interacts with the Chatbot for such indication is outside the scope of this specification and
      - invalidate the corresponding token in the repository and
      - If there is an established 1-to-1 Chat session between the user and the Chatbot where privacy was requested, close the session by sending to the client a SIP BYE request 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").

c) shall send a SIP response according to section 3.6.8.2 with the following clarification:

   i. a ‘tk’ URI parameter (as defined in section 2.5.4.3) set to ‘off’ shall be added to each of the URIs in the P-Asserted-Identity header field.

4. if the request is a SIP OPTIONS or SIP SUBSCRIBE, then the Chatbot Platform shall either respond itself, or interact with the Chatbot without providing the user’s URI.
5. Otherwise (i.e. the conditions in steps 2, 3 and 4 do not apply), the AF

   a) shall provide the user’s URI(s) to the Chatbot as the user’s identity(ies) and
   b) shall send a SIP response according to section 3.6.8.2 with the following clarification:

      i. a ‘tk’ URI parameter (as defined in section 2.5.4.3) set to ‘off’ shall be added to each of the URIs in the P-Asserted-Identity header field.

#### 3.6.5.1.4.2 Processing requests from Chatbots

How the Chatbot platform interacts with the Chatbots and the internal Anonymization Function is up to the implementation.

The AF shall reflect in the P-Asserted-Identity of outgoing SIP request, the identity that was used to interact with the Chatbot:

- If the Chatbot was using a user’s URI, the Chatbot Platform AF shall set the P-Asserted-Identity to the Chatbot URI with a ‘tk’ URI parameter set to ‘off’
- If the Chatbot was using a token, the Chatbot Platform AF shall set the P-Asserted-Identity to the Chatbot URI with a ‘tk’ URI parameter set to ‘on’.

#### 3.6.5.1.4.3 Processing a token deletion

Upon receiving a SIP MESSAGE as defined in section 3.6.5.1.2.3, the Chatbot Platform AF:

1. shall retrieve the URIs of the user and the Chatbot respectively from the P-Asserted-Identity and Request-URI headers;
2. shall check in its repository if a token is associated with the retrieved user and Chatbot URIs;

    a) If there is no association, the Chatbot Platform AF shall return a SIP 200 OK to the client to the received SIP MESSAGE request and shall not proceed with the rest of the steps
    b) otherwise, if there is such association the AF shall invalidate the token and shall return a SIP 200 OK to the client to the received SIP MESSAGE request.
    c) if for any reason, the Chatbot Platform AF cannot access its repository, the Chatbot Platform AF shall respond to the SIP MESSAGE with a SIP 500 Server Internal Error response with optionally a Retry-After header set to a locally configured value and shall not proceed with the rest of the steps.

3. If there is an established 1-to-1 Chat session between the user and the Chatbot where privacy was requested, the AF shall close the session by sending to the client a SIP BYE request 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").

### 3.6.6 Spam and other Inappropriate Chatbot Behaviour Handling

Protection against spam, fraud, inappropriate content or other inappropriate behaviour originating from Chatbots can be realized by preventing access from and to such Chatbots

- in the Spam Block function in the Chatbot Platform or
- in the Spam Block function in the Service Provider network or
- in the client.

Where this protection is performed depends mostly on which party considers the Chatbot as behaving inappropriately. The exception would be Service Providers that can make use of their capabilities for managing the client to inform a client of the Chatbots that they consider problematic. This could be used either in addition to preventing the access in the Service Provider network or as an alternative.

Preventing access to a Chatbot in either the Chatbot Platform or the Service Provider network should affect both the regular messaging services and the retrieval of Chatbot Information (see section 3.6.4). In addition, if the Service Provider blocks the traffic, the Service Provider shall ensure that the corresponding Chatbot cannot be discovered through the Service Provider Chatbot Directory function (see section 3.6.3.1).

If access to a Chatbot is prevented in a Spam Block function, either in the Service Provider network or in the Chatbot Platform, this shall lead to the following behaviour:

- A SIP 403 Forbidden response including a Warning header field including the warning text set to "206 Spam Sender" shall be returned to a SIP request initiated by the client.

- A SIP 404 Not Found response shall be returned to a SIP request initiated by the Chatbot Platform (if blocked in the MNO network).

How this behaviour is realised is network internal and is therefore out of scope of this document. However, RCS Clients receiving such error responses shall inform the user accordingly.

A user shall be able to block the reception of incoming Chatbot communication sessions and messages through the client local blacklist as defined in section 18 of [PRD-RCC.71].

A Service Provider shall be able to prevent access to Chatbots on the client for both incoming and outgoing communication based on the management of blocked Chatbot service identifiers (see section 2.5.4.1) via the Service Provider managed client based spam prevention as described in section 3.6.6.1.

#### 3.6.6.1 Service Provider Managed Client-based Spam Prevention and Prevention of Other Inappropriate Chatbot Behaviour

If the Service Provider wishes to prevent spam, fraud, reception of inappropriate content and other reported inappropriate Chatbot behaviour through the client, the Service Provider shall do so by defining a list of Blacklisted Chatbots via the procedures defined in section 3.6.3.3. The list of Blacklisted Chatbots shall be identified as BLACKLISTED (see <name of the list> in section 3.6.3.3). The specific marker to identify the list shall therefore be:

```
LIST:BLACKLISTED<CRLF>
<CRLF>
```

For every incoming SIP INVITE or SIP MESSAGE request where the P-Asserted-Identity carries an address that corresponds to a Chatbot Service ID, the client shall return a SIP 486 Busy Here response if that Service ID matches an entry in the Blacklisted Chatbots list.

When performing a Service Provider Chatbot Directory lookup as described in section 3.6.3.1, the client shall not show any entries in the result to the user for which the Service ID matches an entry in the Blacklisted Chatbots list.

When the user tries to access a Chatbot (e.g. entered directly by the user, discovered on a website, continuing a conversation started before the Chatbot's Service ID was included on the list) for which the Service ID matches an entry in the Blacklisted Chatbots list , the client shall show an error message to the user without retrieving the Chatbot Information as described in section 3.6.4.

#### 3.6.6.2 Spam Report Message

##### 3.6.6.2.1 Overview

The Spam Report Message allows a client to report a Chatbot as a spammer or as a Chatbot involved in fraud, sending inappropriate content, or other inappropriate behaviour.

The Spam Report Messages are system messages and as such:

- they are not sent with CPIM headers, and a delivery and/or displayed notification shall not be requested,
- there is no store and forward.

The Spam Report Messages are conveyed in a SIP MESSAGE request. They are targeted to the Chatbot address that the user is reporting on and shall only be sent by RCS clients.

To categorize them among other system messages, Spam Report Messages shall use the following IARI:

*+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.botspam"*

The Service Provider may trigger (based on this IARI) an element in its network to process the Spam Report Message, but in any case, the Chatbot Platform hosting the targeted Chatbot shall receive the Spam Report Message.

As for other SIP requests, the Spam Report Message shall be routed towards the AF (if deployed in the Service Provider network).

The Spam Report XML schema is defined as shown in Table 61.

The associated MIME content type is:

*application/vnd.gsma.rcsspam-report+xml*

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:gsma:params:xml:ns:rcs:rcs:spamreport"
                xmlns:xs="http://www.w3.org/2001/XMLSchema"
                xmlns="urn:gsma:params:xml:ns:rcs:rcs:spamreport "
                elementFormDefault="qualified"
                attributeFormDefault="unqualified">
        <xs:element name="SR">
                <xs:complexType>
                        <xs:element name="Chatbot">
                                <xs:simpleType>
                                        <xs:restriction base="xs:anyURI"/>
                                </xs:simpleType>
                        </xs:element>
                        <xs:element name="Message-ID" minOccurs="0" maxOccurs="10">
                                <xs:simpleType>
                                        <xs:restriction base="xs:token"/>
                                </xs:simpleType>
                        </xs:element>
                        <xs:any namespace="##other" processContents="lax" minOccurs="0"
                        maxOccurs="unbounded"/>
                </xs:complexType>
        </xs:element>
</xs:schema>
```

**Table 61: RCS Spam Report Message body schema**

The schema defined in Table 61 is extended to enable the client to include a spam type and free text where the user can explain the reason for classifying the Chatbot message as inappropriate in some way.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:gsma:params:xml:ns:rcs:rcs:spamreportext"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns="urn:gsma:params:xml:ns:rcs:rcs:spamreportext"
        elementFormDefault="qualified"
        attributeFormDefault="unqualified">
        <xs:element name="spam-type" type="xs:string"/>
        <xs:element name="free-text" type="xs:string"/>
</xs:schema>
```

**Table 62: RCS Spam Report Message body schema Extension**

The "spam-type" element shall be added as an extension to the "SR" element of the Spam Report Message Body schema defined in Table 61.

The "free-text" element shall be added as an extension to the "SR" element of the Spam Report Message Body schema defined in Table 61.

#### 3.6.6.2.2 Client Procedures

##### 3.6.6.2.2.1 Sending a Spam Report Message

When a Spam Report Message directed to a specific Chatbot is sent, the client shall send a SIP MESSAGE request according to the rules and procedures of [RCS-CPM-CONVFUNC-ENDORS] with the clarifications listed here. In this SIP MESSAGE request, the client:

1. shall set the Request-URI and To header field to the address of the Chatbot;
2. shall include the address of the originating RCS client that has been authenticated as per [RCS-CPM-CONVFUNC-ENDORS];
3. shall set the P-Preferred-Service header field with the value set to ‘urn:urn-7:3gpp-service.ims.icsi.oma.cpm.systemmsg’;
4. shall include an Accept-Contact header field with the CPM Feature Tag ‘urn:urn-7:3gpp-service.ims.icsi.oma.cpm.systemmsg’ and carrying the Spam Report Message IARI tag defined in section 3.6.6.2.1;
5. shall include a User-Agent header field as specified in Annex C.4.1;
6. shall include the Content-Type header field with the value set to the Spam Report Message content-type *application/vnd.gsma.rcsspam-report+xml*, as described in section 3.6.6.2.1;
7. shall create a Spam Report Message as described in section 3.6.6.2.1 and set the body of the Spam Report Message, as follows:

    a) The <Chatbot> element set to the address of the Chatbot;
    b) Zero or more, up to 10, <Message-ID> elements set to the value of the imdn.message-ID(s) of the original message(s) that is(are) requested to be reported as spam if any;
    c) Optionally the <spam-type> element set to “spam”, “fraud”, “inappropriate-content”, or “other”. If this element is not included it shall be handled as if it had been included with the value “spam”;
    d) Optionally the <free-text> element with text of up to 500 octets.

If the client receives in result of processing the request for the Spam Report SIP MESSAGE request:

- any error, then the client shall notify the user that their request has not been processed.

### 3.6.7 Traffic identification

This section defines the traffic identification CPIM header. Section 3.6.8.7 defines when it is set by the Chatbot Platform and client.

#### 3.6.7.1 Messaging as a Platform CPIM Namespace

A new CPIM namespace is defined for new Messaging as a Platform (MaaP) related CPIM headers.

As per CPIM [RFC3862], this specification defines a new namespace for the CPIM extension header fields defined in the following sections.

The namespace is:

<http://www.gsma.com/rcs/maap>

NOTE: The namespace is considered as a placeholder for a final one to be defined by GSMA or other committees.

As per CPIM [RFC3862] requirements, the new header fields defined in the following sections are prepended, in CPIM messages, by a prefix assigned to the URN through the NS header field of the CPIM message.

The remainder of this specification always assumes an NS header field like this one:

NS: maap <http://www.gsma.com/rcs/maap/>

As specified in [RFC5438], clients are free to use any namespace prefix while servers and intermediaries must accept any legal namespace prefix specification.

#### 3.6.7.2 New CPIM header Traffic-Type

The header is defined as an extension to the [RFC3862] field definitions. The limits for the occurrence of the field are defined in the following table:

| Field | Min Number | Max Number |
|---|---|---|
| Traffic-Type | 0 | 1 |

**Table 63: Traffic-Type header**

The field itself is defined in ABNF as follows. By including the Token, the list of traffic-type values may be extended in a later version of this specification:

```
traffic-type = "Traffic-Type:" traffic-type-value CRLF
traffic-type-value = "advertisement" | "payment" | "premium" |
"subscription" | "plugin" | Token
```

An example CPIM header is `maap.Traffic-Type: advertisement`.

### 3.6.8 Chatbot Service

The Chatbot Service is the service used for the communication between clients and Chatbots. It is based on the RCS Messaging Service.

NOTE: There can be 2 active sessions at a time between a client and a Chatbot: an anonymous session and a non-anonymous session.

When a Chatbot is willing to establish a messaging communication with a client, the Chatbot Platform performs capability discovery towards clients to be contacted. If capability discovery is disabled in the network which the client registered, the network shall follow the requirements described in section 2.6.1.4.2.

#### 3.6.8.1 Chatbot Platform: Initiating a 1-to-1 Chatbot conversation to a user

When initiating communication to a client, a Chatbot Platform shall select either the procedure defined in section 3.6.8.1.1 or the procedure defined in section 3.6.8.1.2 depending on the Chatbot Communication Services that it supports and the services that the client has indicated support for in the Capability Exchange. If both the client and the Chatbot Platform support both the Chatbot Chat Session and the Chatbot Standalone Messaging technologies, the Chatbot Platform shall initiate a Chatbot conversation to a user based on the Chatbot's use cases. Chatbot Chat Sessions should for example be used if

- The Chatbot “knows” that a conversational exchange is expected
- towards anonymised contacts

##### 3.6.8.1.1 Chatbot Platform: Initiating a 1-to-1 Chatbot Chat Session request to a user

When a request from a Chatbot to initiate a 1-to-1 Chatbot conversation towards a user is to be sent and the Chatbot Chat Session service is selected, the Chatbot Platform shall send a SIP INVITE request according to section 3.2.3.1 of this specification and to the rules and procedures of section 7.3.1.1 of [RCS-CPM-CONVFUNC-ENDORS] with the clarifications listed here.

In this SIP INVITE request, the Chatbot Platform:

1. shall include the Chatbot IARI, the Chatbot application version feature tag and the *isbot* feature tag as defined in section 3.6.2, in the Contact header in addition to values already included following section 3.2.3.1;
2. shall in addition to the Accept-Contact header already added, add another Accept-Contact header field carrying the Chatbot IARI feature tag defined in section 3.6.2.1 and the Chatbot application version feature tag defined in section 3.6.2.2 and shall include the *require* and *explicit* parameters;
3. shall set the P-Asserted-Service header field with the value of the CPM Feature Tag ‘urn:urn-7:3gpp-service.ims.icsi.oma.cpm.session’;
4. shall set the P-Asserted-Identity header to the service-ids of the Chatbot defined as per section 2.5.4.1;
5. shall, in addition to values to be included as per section 3.2.3.1, include in the a=accept-wrapped-types the list of the Chatbot-related content-types defined in section 3.6.10.2 that the Chatbot Platform is willing to receive, i.e.

*application/vnd.gsma.botsuggestion.response.v1.0+json* and *application/vnd.gsma.botsharedclientdata.v1.0+json*.

Upon receipt of a SIP response, section 3.2.3.1 and the rules and procedures of section 7.3.2 of [RCS-CPM-CONVFUNC-ENDORS] apply. If a 1-to-1 Chatbot Chat session is set up and the terminating network indicates support for revocation as per section 3.2.3.8.2, the Chatbot Platform may inform the Chatbot that it may revoke Chatbot messages if required.

Even though the Chatbot application version feature tag is returned in the Contact header of the SIP INVITE response, the Chatbot Platform shall use the value received via capability exchange. If the served user is offline, the Messaging Server may not know the actual version supported by the user's device.

Non-normative example:

- Contact header of a SIP INVITE request sent from the Chatbot Platform:

```
Contact:<sip:foo.bar@botplatform.domain>;+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.oma.cpm.session";+g.3gpp.iari-ref="urn%3Aurn-7%3A3gpp-application.ims.iari.rcs.chatbot";+g.gsma.rcs.botversion="#=1";+g.gsma.rcs.isbot
```

#### 3.6.8.1.2 Chatbot Platform: Initiating a 1-to-1 Chatbot Standalone Message request to a user

When a request from a Chatbot to initiate a 1-to-1 Chatbot Conversion towards a user is to be sent, and the Chatbot Standalone Message is selected, the Chatbot Platform shall send a SIP MESSAGE request according to section 3.2.2.1 of this specification and to the rules and procedures of section 7.2.1.1 of [RCS-CPM-CONVFUNC-ENDORS] with the clarifications listed here.

In this SIP MESSAGE request, the Chatbot Platform:

1. shall include the isbot and Chatbot application version feature tags as defined in section 3.6.2, in the CPIM From header in addition to values already included following section 3.2.2.6;
2. shall in addition to the Accept-Contact header already added, add another Accept-Contact header field carrying the Chatbot IARI feature tag defined in section 3.6.2.1 and the Chatbot application version feature tag defined in section 3.6.2.2 and shall include the *require* and *explicit* parameters;
3. shall set the P-Asserted-Service header field with the value of the CPM Feature Tag 'urn:urn-7:3gpp-service.ims.icsi.oma.cpm.msg'
4. shall set the P-Asserted-Identity header to the service-ids of the Chatbot defined as per section 2.5.4.1;

Non-normative example:

- CPIM.From of a SIP MESSAGE request sent from the Chatbot Platform:

```
From:
<sip:anonymous@anonymous.invalid?Contact=+g.gsma.rcs.isbot%3B+g.gsma.
rcs.botversion%3D%22%231%22>
```

Upon receipt of a SIP response, section 3.2.2.1 and the rules and procedures of section 7.2.2 of [RCS-CPM-CONVFUNC-ENDORS] apply.

#### 3.6.8.2 Chatbot Platform: Receiving a 1-to-1 Chatbot conversation request from a user

##### 3.6.8.2.1 Chatbot Platform: Receiving a 1-to-1 Chatbot Chat Session request from a user

When handling a request from a user to set up a 1-to-1 Chatbot Chat Session with a Chatbot, section 3.2.3.1 and the rules and procedures of section 7.3.2 of [RCS-CPM-CONVFUNC-ENDORS] apply as well as the clarifications listed here.

In this SIP INVITE response, the Chatbot Platform:

1. shall include the Chatbot IARI, the Chatbot application version feature tag and the isbot feature tag as defined in section 3.6.2, in the Contact header in addition to values already included following section 3.2.3.1.
2. shall set the P-Asserted-Identity header to the service-ids of the Chatbot defined as per section 2.5.4.1
3. shall, in addition to values to be included as per section 3.2.3.1, include in the a=accept-wrapped-types the list of the Chatbot-related content-types defined in section 3.6.10.2 that the Chatbot Platform is willing to receive, i.e. *application/vnd.gsma.botsuggestion.response.v1.0+json* and *application/vnd.gsma.botsharedclientdata.v1.0+json*.

When handling a request from a user to set up a 1-to-1 Chat Session with a Chatbot, and a Chatbot IARI is not included in the SIP INVITE request, the Chatbot Platform shall return a SIP 403 Forbidden response code including a warning header set to “488 Chatbot Conversation Needed” to the user.

Since there is no requirement for the Chatbot Platform to provide delivery assurance for messages from a user to the Chatbot, section 3.2.3.8 does not apply and the Chatbot Platform shall not indicate NFS and shall not indicate revocation support in its 200 OK response to the incoming SIP INVITE request.

##### 3.6.8.2.2 Chatbot Platform: Receiving a 1-to-1 Chatbot Standalone Message request from a user

When handling a SIP MESSAGE request with the OMA CPM ICSI “3gpp-service.ims.icsi.oma.cpm.msg” from a user, section 3.2.2.1 and the rules and procedures of section 7.2.2 of [RCS-CPM-CONVFUNC-ENDORS] apply as well as the clarifications listed here.

In this SIP MESSAGE response, the Chatbot Platform:

1. shall set the P-Asserted-Identity header to the service-ids of the Chatbot defined as per section 2.5.4.1

When handling a request from a user to set up a 1-to-1 Standalone Message with a Chatbot without Chatbot IARI in the SIP MESSAGE request:

- If follow-up conversation from the user is expected by the Chatbot, the Chatbot Platform Shall return a SIP 403 Forbidden response code including a warning header set to “488 Chatbot Conversation Needed” to the user.
- If follow-up conversation from the user is not expected by the Chatbot, the Chatbot Platform Shall return a SIP 200 OK response.

NOTE: the client can know the contact is a Chatbot from the P-Asserted-Identity header of SIP 200 OK response.

#### 3.6.8.2.3 Handling incoming requests for a temporarily unavailable Chatbot

A Chatbot platform may support handling of Chatbots to be temporarily unavailable; the Chatbots can be suspended or temporarily out of service.

When receiving a capability query from a user to a temporarily unavailable Chatbot, the Chatbot Platform shall return the correct capabilities to the user as per section 3.6.2.4.

When receiving a Chatbot Chat Session initiation request as defined in section 3.6.8.3 from a user to a temporarily unavailable Chatbot, the Chatbot Platform:

1. shall accept the session as per the procedures defined in section 3.6.8.2;
2. shall send a Chatbot specific “out of service” message in the MSRP session;
3. shall close the session by sending a SIP BYE request.

When receiving a Chatbot Standalone Message request as defined in section 3.6.8.3 from a user to a temporarily unavailable Chatbot, the Chatbot Platform:

1. shall send a Chatbot specific “out of service” message in a Pager Mode Standalone Message toward the user using the procedure defined in section 3.6.8.1.2;

#### 3.6.8.2.4 Handling incoming requests for a restricted Chatbot

When a user attempts to contact a Chatbot, and the Chatbot access is restricted for the user (for example, a Chatbot providing services only to users of a specific MNO) the Chatbot Platform shall respond to the SIP request or Capability request with a SIP 403 Forbidden response with a Warning header field including the warning text set to “205 Chatbot has declined”.

### 3.6.8.3 Client: Initiating a 1-to-1 Chatbot conversation request to a Chatbot

When initiating communication to a Chatbot, the client shall select the procedure defined in section 3.6.8.3.1, the procedure defined in section 3.6.8.3.2 or SMS as defined in section 3.2.1.2.

#### 3.6.8.3.1 Client: Initiating a 1-to-1 Chatbot Chat Session request to a Chatbot

When a request from a user’s client to initiate a 1-to-1 Chatbot Chat Session towards a Chatbot is to be sent, the client shall send a SIP INVITE request according to section 3.2.3.1 and to the rules and procedures of section 7.3.1.1 of [RCS-CPM-CONVFUNC-ENDORS] with the clarifications listed here.

In this SIP INVITE request, the client:

1. shall in addition to values already included in the Contact header, include the Chatbot IARI and the Chatbot application version feature tag as defined in section 3.6.2, in the Contact header;
2. shall in addition to the Accept-Contact header already added, add another Accept-Contact header field carrying the Chatbot IARI feature tag defined in section 3.6.2.1 and the Chatbot application version feature tag defined in section 3.6.2.2, and shall include the *require* and *explicit* parameters;
3. shall set the Request-URI of the Chat session request to the service-id of the Chatbot defined as per section 2.5.4.1. The Chatbot SIP URI should be used if available instead of the tel URI;
4. shall, in addition to values to be included as per section 3.2.3.1, include in the a=accept-wrapped-types the list of the Chatbot-related content-types defined in 3.6.10.2 that the client is willing to receive, i.e. multipart/mixed, application/vnd.gsma.botsuggestion.v1.0+json, and application/vnd.gsma.botmessage.v1.0+json.
5. shall, when privacy is requested for the conversation, add a privacy header as defined in [RFC3323] if not already included, and add a value “tk” to it.
6. shall, when initiating a session to link the user's token to their actual identity, add a privacy header as defined in [RFC3323] if not included already and add a value “tklink” to it.

Upon receipt of a SIP 200 OK response,

- the client shall verify whether the Contact header of the SIP 200 OK response contains the Chatbot role feature tag as defined in section 3.6.2.3.
- If the Chatbot role feature tag is not provided, the client shall not establish the media plane and terminate the SIP Session by sending a SIP BYE request to the Chatbot according to the rules and procedures of section 7.3.4.1 of [RCS-CPM-CONVFUNC-ENDORS]
- Otherwise, section 3.2.3.1 and the rules and procedures of section 7.3.2 of [RCS-CPM-CONVFUNC-ENDORS] apply. There shall be no indication of support for delivery assurance by the Chatbot Platform in the SIP response.

If receiving a SIP final response different from SIP 200 OK, the rules and procedures of section 7.3.1.1 of [RCS-CPM-CONVFUNC-ENDORS] apply

Upon receipt of a SIP 403 Forbidden error response with a Warning header with the warning text set to “204 Token Not Found” to a Chatbot Chat Session initiation requesting to link the user's token to their actual identity, the client shall consider that it is not possible to link an existing token for the user with their MSISDN, and the client shall not automatically retry the token link request. In this case the client shall check if there is an active session with that particular Chatbot; if there is one and it is anonymised, the client shall close that session.

Non-normative example:

- Contact header of a SIP INVITE request sent from the client:

```
Contact:<sip:foo.bar@domain>;+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-
service.ims.icsi.oma.cpm.session";+g.3gpp.iari-ref="urn%3Aurn-
7%3A3gpp-
application.ims.iari.rcs.chatbot";+g.gsma.rcs.botversion="#=1"
```

#### 3.6.8.3.2 Client: Initiating a 1-to-1 Chatbot Standalone Message request to a Chatbot

When a client needs to send a 1-to-1 Chatbot Standalone Message towards a Chatbot, the client shall send a SIP MESSAGE request according to section 3.2.2.1 of this specification and to the rules and procedures of section 7.2.1.1 of [RCS-CPM-CONVFUNC-ENDORS] with the clarifications listed here.

In the SIP MESSAGE request, the client:

1. shall in addition to the Accept-Contact header already added, add another Accept-Contact header field carrying the Chatbot IARI feature tag defined in section 3.6.2.1 and the Chatbot application version feature tag defined in section 3.6.2.2, and shall include the require and explicit parameters;
2. shall set the Request-URI of the SIP MESSAGE request to the service-id of the Chatbot defined as per section 2.5.4.1. The Chatbot SIP URI should be used if available instead of the tel URI;

Upon receipt of a SIP 200 OK response,

- Section 3.2.2.1 and the rules and procedures of section 7.2.2 of [RCS-CPM-CONVFUNC-ENDORS] apply.

If receiving a SIP final response different from SIP 200 OK, the rules and procedures of section 7.2.1.1 of [RCS-CPM-CONVFUNC-ENDORS] apply.

### 3.6.8.4 Client: Receiving a 1-to-1 Chatbot conversation request from a Chatbot

#### 3.6.8.4.1 Client: Receiving a 1-to-1 Chatbot Chat Session request from a Chatbot

When receiving a SIP INVITE for a Chat session with an Accept-Contact header field containing the Chatbot IARI feature tag defined in section 3.6.2.1, the client

- shall reject the SIP INVITE request with a SIP 488 NOT ACCEPTABLE HERE response if the client is not authorised to use Chatbot Communication using Sessions (see section 3.6.8.9), otherwise
- shall reject the SIP INVITE request with a SIP 606 Not Acceptable response, if the Contact header of the SIP INVITE request does not contain the Chatbot role feature tag as defined in section 3.6.2.3, otherwise
- shall reject the SIP INVITE request with a SIP 606 Not Acceptable response, if privacy is enabled using the PRIVACY DISABLE client configuration parameter defined in section A.1.3 and the P-Asserted-Identity in the SIP INVITE request doesn't contain the 'tk' parameter with a value "on" or "off"; otherwise
- shall apply the rules and procedures of section 3.2.3.1 of this document and of section 7.3.2 of [RCS-CPM-CONVFUNC-ENDORS] with the clarifications listed below:

- shall if privacy is enabled using the PRIVACY DISABLE client configuration parameter defined in section A.1.3, check the value of the 'tk' parameter in the Chatbot URI provided in the P-Asserted-Identity in the SIP INVITE request:
  - If the value of the 'tk' parameter is 'off', the client shall assume that the Chatbot is using the user's identity.
  - Otherwise, the client shall assume that a token is used between the Chatbot and the AF.
- When the client returns a SIP 200 OK response, the client
  - shall include in the Contact header of the SIP 200 OK response the Chatbot IARI as defined in section 3.6.2.1 and the Chatbot application version feature tags as defined in section 3.6.2.2
  - shall, in addition to values to be included as per section 3.2.3.1, include in the a=accept-wrapped-types the list of the Chatbot-related content-types defined in 3.6.10.2 that the client is willing to receive, i.e. *multipart/mixed*, *application/vnd.gsma.botsuggestion.v1.0+json*, and *application/vnd.gsma.botmessage.v1.0+json*.
  - shall check if the Chatbot Information needs to be refreshed according to section 3.6.4.

#### 3.6.8.4.2 Client: Receiving a 1-to-1 Chatbot Standalone Message request from a Chatbot

When receiving a SIP MESSAGE with an Accept-Contact header field containing the Chatbot IARI feature tag defined in section 3.6.2.1 and another Accept-Contact header with the CPM Standalone Messaging ICSI, the client

- shall reject the SIP MESSAGE request with a SIP 488 NOT ACCEPTABLE HERE response if the client is not authorised to use Chatbot Communication using Standalone Messaging (see section 3.6.8.9), otherwise
- shall reject the SIP MESSAGE request with a SIP 606 Not Acceptable response, if the CPIM From header of the SIP MESSAGE does not contain the Chatbot role feature tag as defined in section 3.6.2.3, otherwise
- shall apply the rules and procedures of section 3.2.2.1 of this document and of section 7.2.2 of [RCS-CPM-CONVFUNC-ENDORS] with the clarifications listed below:
  - When the client returns a SIP 200 OK response, the client shall check if the Chatbot Information needs to be refreshed according to section 3.6.4.

### 3.6.8.5 Messaging Server serving a user: handling incoming 1-to-1 conversation requests from a Chatbot

#### 3.6.8.5.1 Messaging Server serving a user: handling incoming 1-to-1 Session requests from a Chatbot

When handling an incoming 1-to-1 chat session request with the Chatbot IARI from a Chatbot the rules and procedures of [RCS-CPM-CONVFUNC-ENDORS] apply as well as the clarifications listed here.

The user's Messaging Server

1. shall indicate in the response as per section 3.2.3.8.1 "Network Fallback Support Capability" whether it supports revocation for chat messages received from a Chatbot.
2. shall in addition to values already to be included in the Contact header in the response, include the Chatbot IARI and the Chatbot application version feature tag as defined in section 3.6.2, in the Contact header. The Chatbot application version shall be set to the version supported by the served user if available, otherwise, shall be set to the version supported by the Messaging Server;
3. shall, in addition to values to be included as per section 3.2.3.1, include in the a=accept-wrapped-types of the response the list of the Chatbot-related content-types defined in 3.6.10.2 that the Messaging Server is willing to receive on behalf of the client. This will match the content-types associated with the supported Chatbot application version of the served user if available, otherwise, shall be set to the content-types supported by the Messaging Server;
4. shall not indicate in the response support for Network Fallback to SMS (NFS) for this session, and shall not perform NFS, even if NFS is supported in general, since the incoming request includes the require and explicit parameters on a dedicated Accept-Contact header field carrying the Chatbot IARI feature tag defined in section 3.6.2.1.
5. shall, if none of the P-Asserted-Identities in the incoming SIP INVITE request contains a 'tk' URI parameter (as defined in section 2.5.4.3), add a 'tk' URI parameter set to 'off' to each URI in the P-Asserted-Identity header before propagating the request towards the served user.

#### 3.6.8.5.2 Messaging Server serving a user: handling incoming 1-to-1 Standalone Message requests from a Chatbot

When handling an incoming SIP MESSAGE request with the Chatbot IARI from a Chatbot, the rules and procedures of [RCS-CPM-CONVFUNC-ENDORS] apply as well as the clarifications listed here.

The user's Messaging Server

1. shall not perform interworking to xMS even if interworking is supported in general, since the incoming request includes the require and explicit parameters on a dedicated Accept-Contact header field carrying the Chatbot IARI feature tag defined in section 3.6.2.1.

When it receives MessageRevoke requests from a Chatbot Platform, the Messaging Server shall handle the request and shall send a MessageRevokeResponse request as described in section 3.2.2.5.2;

#### 3.6.8.6 Messaging Server serving a user: handling outgoing 1-to-1 conversation requests to a Chatbot

##### 3.6.8.6.1 Messaging Server serving a user: handling outgoing 1-to-1 Session requests to a Chatbot

When handling an outgoing 1-to-1 chat session request with the Chatbot IARI from a user, the rules and procedures of [RCS-CPM-CONVFUNC-ENDORS] apply as well as the clarifications listed here.

The user's Messaging Server

1. shall indicate in the request as per section 3.2.3.8.1 "Network Fallback Support Capability" whether it supports revocation for chat messages received from a Chatbot.
2. shall not indicate support for Network Fallback to SMS (NFS) for this session, and shall not perform NFS, even if NFS is supported in general, since the outgoing request includes the *require* and *explicit* parameters in a dedicated Accept-Contact header field carrying the Chatbot IARI feature tag defined in section 3.6.2.1.
3. if a Privacy header is present and contains the value "tk" then for any received responses that contain one or more P-Asserted-Identity headers add a 'tk' URI parameter (as defined in section 2.5.4.3) set to 'on' to each URI in the P-Asserted-Identity header, if not already present; Otherwise,
4. if no Privacy header is present or if a Privacy header is present but it does not contain the value "tk" then for any received responses that contain a P-Asserted-Identity header, add a 'tk' URI parameter (as defined in section 2.5.4.3) set to 'off' to each URI in the P-Asserted-Identity header, if not already present.

There shall be no indication of support for delivery assurance by the Chatbot Platform in the incoming SIP INVITE response.

##### 3.6.8.6.2 Messaging Server serving a user: handling outgoing 1-to-1 Standalone Message requests to a Chatbot

When handling an outgoing 1-to-1 SIP MESSAGE request with the Chatbot IARI from a user, the rules and procedures of [RCS-CPM-CONVFUNC-ENDORS] apply with no additional requirement.

#### 3.6.8.7 Chat messages exchanged within a 1-to-1 Chat session with a Chatbot

Chat messages within the 1-to-1 chat session with a Chatbot are exchanged according to the rules and procedures of section 8.6 of [RCS-CPM-CONVFUNC-ENDORS] apply as well as the clarifications listed here.

If the user's network indicated support for revocation, and if requested by the Chatbot, the Chatbot Platform shall initiate the MessageRevoke procedures as described in section 3.2.3.8.2. The Chatbot Platform shall inform the Chatbot of the result of the message revocation request.

When a Chatbot sends a message of one of the traffic types defined in section 3.6.7.2 the Chatbot Platform shall add the CPIM traffic type header with the corresponding value defined in section 3.6.7.2.

When a client receives a message with any traffic type header value defined in section 3.6.7.2 that includes a suggested reply, the client shall add the same CPIM traffic type header and value in the suggested reply response that it sends on behalf of the user.

The following additional rule applies for the user's Messaging Server when processing CPIM header extensions according to the media plane handling defined in section 2.14.3.2.

If for a 1-to-1 Chatbot session, CPIM header extensions are not supported according to the definitions in section 2.14.3.2, and

1. if the Messaging Server needs to send a message with a "NS" header containing the URI value defined in section 3.6.7.1, then the Messaging Server

   - shall not remove the "NS" header, and
   - shall not remove the CPIM headers containing the associated name prefix,

2. if the Messaging Server needs to send a message with a "NS" header containing the URI value defined in section C.1.13 of [RCS-CPM-CONVFUNC-ENDORS], then the Messaging Server

   - shall not remove the "NS" header, and
   - shall not remove the CPIM headers containing the associated name prefix.

For payload that is sent from the client to the Chatbot platform, there are two possible scenarios:

- If there is a 1-to-1 chat session established between the client and the Chatbot platform, it shall be re-used to convey the payload as defined in section 3.6.10.6.2.
- If there is no 1-to-1 chat session established, the client shall initiate a session based on the procedures described in section 3.6.8.3. Once the session is established, the client shall use it to send the payload as defined in section 3.6.10.6.2.

#### 3.6.8.8 Delivery and Display Notifications

The rules for sending delivery and display notifications shall follow those of section 3.2.2.2 and 3.2.3.1.

In addition, for any disposition notification sent outside of an established MSRP session (i.e. sent by SIP MESSAGE) and related to a message that was received by the client in an anonymous Chatbot Chat Session (i.e. the 'tk' parameter of the Chatbot URI was set to ‘on’ in the P-Asserted-Identity received by the client), the client shall add to the SIP MESSAGE conveying the disposition notification a Privacy header as defined in [RFC3323] if not already included, and add a value “tk” to it.

#### 3.6.8.9 Client: Authorisation of Chatbot Communication Services

The client is authorised to offer the Chatbot Communication Services to the user if the configuration parameter CHATBOT MSG TECH as defined in section A.1.4 is set to a value different from "0". The client is authorised to use Chatbot Communication using Sessions for communication with a Chatbot if the configuration parameter CHATBOT MSG TECH as defined in section A.1.4 is set to "1" or "2" and shall in this case include the capability for Chatbot Communication using Sessions in the Capability Exchange. The client is authorised

to use Chatbot Communication using Standalone Messaging if the configuration parameter CHATBOT MSG TECH as defined in section A.1.4 is set to "2" or "3" and shall in this case include the capability for Chatbot Communication using Standalone Messaging in the Capability Exchange.

### 3.6.9 Deferred Messaging

Chat messages to a user from a Chatbot that cannot be immediately delivered shall be deferred according to the rules and procedures of sections 8.3.1.6.1, 8.3.2.9 and 8.2.2.3 of [RCS-CPM-CONVFUNC-ENDORS] as well as the clarifications listed here. Upon deferral, the Messaging Server:

1. shall ensure that the Accept-Contact header including the Chatbot IARI and the Accept-Contact header including the Chatbot application version and the *require* and *explicit* parameters are kept per deferred message,
2. shall ensure that all CPIM headers including the new Chatbot related ones are stored with the Chatbot message; and
3. shall ensure that separate deferred messaging queues are kept for messages and disposition notifications from the Chatbot sent in a SIP MESSAGE request or a session that was anonymised, and for messages and disposition notifications sent in a SIP MESSAGE request or a session that was not anonymised. The 'tk' URI parameter with value other than “off” on the SIP URI in the P-Asserted-Identity header means the session or SIP MESSAGE request was anonymised, and 'tk=off' means the session or SIP MESSAGE request was not anonymised. The 'tk' URI parameter with its value shall be kept with the SIP URI identifying the deferred messaging queue. See also section 3.6.5.1.

Chat messages to a user from a Chatbot that have been deferred shall be delivered according to the rules and procedures of sections 8.3.1.6.2, 8.3.1.6.6, 8.3.2.9 and 8.3.2.9.1.1 of [RCS-CPM-CONVFUNC-ENDORS] as well as the clarifications listed here. Upon deferred delivery, the Messaging Server:

1. for messages and disposition notifications from one or more Chatbot sessions or SIP MESSAGE requests that were anonymised (i.e. deferred P-Asserted-Identity has a SIP URI with the 'tk' URI parameter set to a value other than “off”)

    a) shall for a session include in the Contact header the Chatbot IARI, the Chatbot application version feature tag and the isbot feature tag as defined in section 3.6.2 set to the value that was deferred with the chat messages;
    b) shall for a message sent in a SIP MESSAGE request include the isbot feature tag in the CPIM From header as defined in section 3.2.2.1;
    c) shall add to the Accept-Contact header already added the Chatbot IARI feature tag defined in section 3.6.2.1 and add another Accept-Contact header with the Chatbot application version feature tag defined in section 3.6.2.2 set to the value that was deferred with the chat messages and shall include the *require* and *explicit* parameters;
    d) shall set the Referred-By header for a session and the P-Asserted-Identity header for a SIP MESSAGE request to the SIP URI service-id of the Chatbot defined as per section 2.5.4.1 that was deferred in the P-Asserted-Identity header with the chat messages, including the 'tk' URI parameter and its value,

and may include the tel URI service-id of the Chatbot if it was also deferred with the chat messages;

e) Shall for a session, in addition to values to be included as per section 3.2.3.1, include in the a=accept-wrapped-types the list of the Chatbot-related content-types defined in section 3.6.10.2 that the Chatbot Platform is willing to receive.

And

2. for messages from one or more Chatbot sessions or SIP MESSAGE requests that were not anonymised (i.e. deferred P-Asserted-Identity has a SIP URI with the 'tk=off' URI parameter)

    a) repeat steps 1.a) to 1.e) above in this section.

NOTE: Because this procedure is the same as in earlier versions, a Chatbot Platform will not have to support the special session for the delivery of stored IMDNs described in section 3.2.3.3.

Upon receipt of a SIP response, the rules and procedures of section 8.3.2.9.1.1 of [RCS-CPM-CONVFUNC-ENDORS] apply for session setup and for delivering the deferred Chat messages.

Example when two values are carried in the Referred-By header:

```
Referred-By:<sip:boty@botplatform.botplatformz.com;tk=on>;add-
refs="tel:+123456789;tk=on"
```

### 3.6.10 Rich Cards and Suggested Chip Lists

#### 3.6.10.1 Payloads

Chatbots can send two different types of messages to RCS clients:

- Regular RCS messages using existing content types (e. g. plain text, file transfer, or geo location push), or
- Chatbot messages using the content type as defined in section 3.6.10.2.1, currently only used for sending Rich Cards

NOTE: Due to the inclusion of multiple HTTP links, postback data, etc. in the schema of Rich Cards and Chip lists, the size of the JSON payload may be much larger than for conventional person-to-person Chat. It is suggested that developers, Chatbot Platforms and MNOs assume a maximum size of 250KB for these JSON payloads.

Both message types can either be

- combined with a Suggested Chip List or
- sent on their own without a Suggested Chip List.

Whenever messages are combined with a Suggested Chip List, they will be sent as a multipart CPIM message. Only combining messages and Suggested Chip List into a

multipart CPIM message is allowed. Chatbot Platforms shall not combine more than one message (e.g. a Rich Card and a plain text message) in a multipart CPIM message. A Suggested Chip List shall not be sent without a message to which it is to be associated.

Whenever messages are sent on their own without a Suggested Chip List they will not be sent as a multipart CPIM message but as a regular, single CPIM message.

The benefit of sending message and Suggested Chip List in a single, multipart CPIM message is the atomicity. This prevents clients from entering an undefined state where only one of the CPIM messages reaches the client. It also provides a user experience where both message and suggestions are always shown at the same time.

Multipart CPIM messages shall include a filename parameter in the Content-Disposition header for each part of the Multipart message, to allow for correct storage as per [CPM-MSGSTOR-REST]. Each part shall have a unique filename and these shall be added by the Chatbot Platform.

The following table provides an overview of when to use multipart CPIM:

| | with Suggested Chip List | without Suggested Chip List |
|---|---|---|
| **Regular RCS message** | multipart CPIM | single CPIM |
| **Chatbot message** | multipart CPIM | single CPIM |

**Table 64: use of multipart bodies**

When receiving a non-conformant message (e.g. non-compliant to the JSON or XML schema or a multipart message with inappropriate body types), the client shall acknowledge the reception and send a delivery notification, but shall not display the message to the user.

NOTE: Chatbot Platforms are expected to verify the content that is sent and as such, this case should not occur in practice. This silent discarding is therefore not considered a problem.

NOTE: This behaviour might be used also when receiving non-conformant messages in a regular person-to-person communication.

### 3.6.10.2 Content Types

#### 3.6.10.2.1 Chatbot Message for Rich Cards

A Chatbot Message containing a Single Rich Card or a Carousel of Rich Cards must be wrapped as a CPIM message in a Chat message with the following content type:

`application/vnd.gsma.botmessage.v1.0+json`

Note: Chatbots can also send regular RCS messages as stated in section 3.6.10.1 above.

#### 3.6.10.2.2 Suggested Chip List

A suggestion (suggested replies and suggested actions) must be wrapped as a part of a multipart CPIM message with the following content type:

```
application/vnd.gsma.botsuggestion.v1.0+json
```

#### 3.6.10.2.3 Client Response to Suggestion

A response from an RCS client based on the suggestions provided must be wrapped as a CPIM message in a chat message with the following content type:

```
application/vnd.gsma.botsuggestion.response.v1.0+json
```

#### 3.6.10.2.4 Data shared by the client to the Chatbot

As a result of the user interacting with certain suggested actions, clients can share data with Chatbots. This data must be wrapped as a CPIM message in a chat message with the following content type:

```
application/vnd.gsma.botsharedclientdata.v1.0+json
```

#### 3.6.10.2.5 SDP Content

The `a=accept-wrapped-types` attribute must be included by Chatbot Platforms and clients for the content types defined above for Chatbot Messages, Suggested Chip Lists and responses. The content-types to be included by Chatbot Platforms are listed in section 3.6.8.1, and the content-types to be included by clients are listed in section 3.6.8.3. These content-types are always sent as part of a CPIM message and therefore should not be included in the `a=accept-types` attribute.

Example of the SDP offer/answer for a 1-to-1 Chatbot Chat Session sent from client to Chatbot:

```
Content-Type: application/sdp

v=0
o=- 3688370227 3688370227 IN IP4 47.73.238.12
s=
c=IN IP4 10.100.1.2
t=0 0
m=message 13500 TCP/MSRP *
a=accept-types:message/cpim application/im-iscomposing+xml
a=path:msrp://10.100.1.2:13500/a7f2d26995f6c206;tcp
a=sendrecv
a=accept-wrapped-types:message/imdn+xml
application/vnd.gsma.rcspushlocation+xml text/plain
application/vnd.gsma.rcs-ft-http+xml multipart/mixed
application/vnd.gsma.botsuggestion.v1.0+json
application/vnd.gsma.botmessage.v1.0+json
a=setup:active
```

**Table 65: Chatbot Communication SDP example**

#### 3.6.10.2.6 Example of a multipart CPIM message

The following example contains a File Transfer via HTTP XML message, combined with a Chatbot Suggested Chip List:

```
Content-Type: message/cpim
```

```
From: <sip:anonymous@anonymous.invalid>
To: <sip:anonymous@anonymous.invalid>
DateTime: 2016-11-17T11:17:08.589Z
NS: imdn <urn:ietf:params:imdn>
NS: cpm <http://www.openmobilealliance.org/cpm/>
cpm.Payload-Type:application/vnd.gsma.rcs-ft-
http+xml;application/vnd.gsma.botsuggestion.v1.0+json
imdn.Message-ID: JOBlDiOnVaK9FpQTeOasX6wGptTjkvD6
imdn.Disposition-Notification: positive-delivery, display

Content-Type: multipart/mixed; boundary="next"

--next
Content-Type: application/vnd.gsma.rcs-ft-http+xml
Content-Disposition: attachment; filename="DSC_379395051.JPG"
Content-Length: [content length]

<?xml version="1.0" encoding="UTF-8"?>
<file>
 <file-info type="thumbnail">
   <file-size>7427</file-size>
   <content-type>image/jpeg</content-type>
   <data
url="https://ftcontentserver.rcs.mnc123.mcc456.pub.3gppnetwork.org/ftsf-
58cdb29d1-a3d3-427c-a8a4-a496759fbf6b" until="2017-04-25T12:17:07Z"/>
 </file-info>
  <file-info type="file">
   <file-size>183524</file-size>
   <file-name>DSC_379395051.JPG</file-name>
   <content-type>image/jpeg</content-type>
   <data
url="https://ftcontentserver.rcs.mnc123.mcc456.pub.3gppnetwork.org/ftsf-
0d5ea6d1-a94c-2-9634-2d90244d3e8e" until="2017-04-25T12:17:07Z"/>
 </file-info>
</file>
--next
Content-Type: application/vnd.gsma.botsuggestion.v1.0+json
Content-Disposition: attachment; filename="Chiplist.lst"
Content-Length: [content length]

{
  "suggestions": [
    {
      "reply": {
        "displayText": "Yes",
        "postback": {
          "data": "set_by_chatbot_reply_yes"
        }
      }
    },
    {
      "reply": {
        "displayText": "No",
```

```
        "postback": {
          "data": "set_by_chatbot_reply_no"
        }
      }
    },
    {
      "action": {
        "urlAction": {
          "openUrl": {
            "url": "https://www.google.com"
          }
        },
        "displayText": "Open website or deep link",
        "postback": {
          "data": "set_by_chatbot_open_url"
        }
      }
    },
    {
      "action": {
        "dialerAction": {
          "dialPhoneNumber": {
            "phoneNumber": "+1650253000"
          }
        },
        "displayText": "Call a phone number",
        "postback": {
          "data": "set_by_chatbot_open_dialer"
        }
      }
    }
  ]
}
--next--
```

**Table 66: Chatbot Communication multipart CPIM message**

#### 3.6.10.2.7 Text Encoding in JSON payload

All fields within JSON payloads containing text should be encoded in UTF-8.

#### 3.6.10.2.8 Date/Time Format in JSON payload

All date/time fields within JSON payloads should be specified as defined in ISO 8601, including all three fields, date, time, and time zone offset.

### 3.6.10.3 Data structure overview

Figure 14 shows a high-level overview of the data structure for all JSON payloads, each containing one of:

- message, which is a Chatbot Message sent from Chatbot Platform to client,
- suggestions, which is a Suggested Chip List sent from Chatbot Platform to client, or
- response, which is a response sent from client to Chatbot Platform,
- sharedData, which is Shared Client Data sent from client to Chatbot Platform.

Note: This overview does not contain all fields and possible sub-objects. See the JSON schema in section 3.6.10.4 for more details.

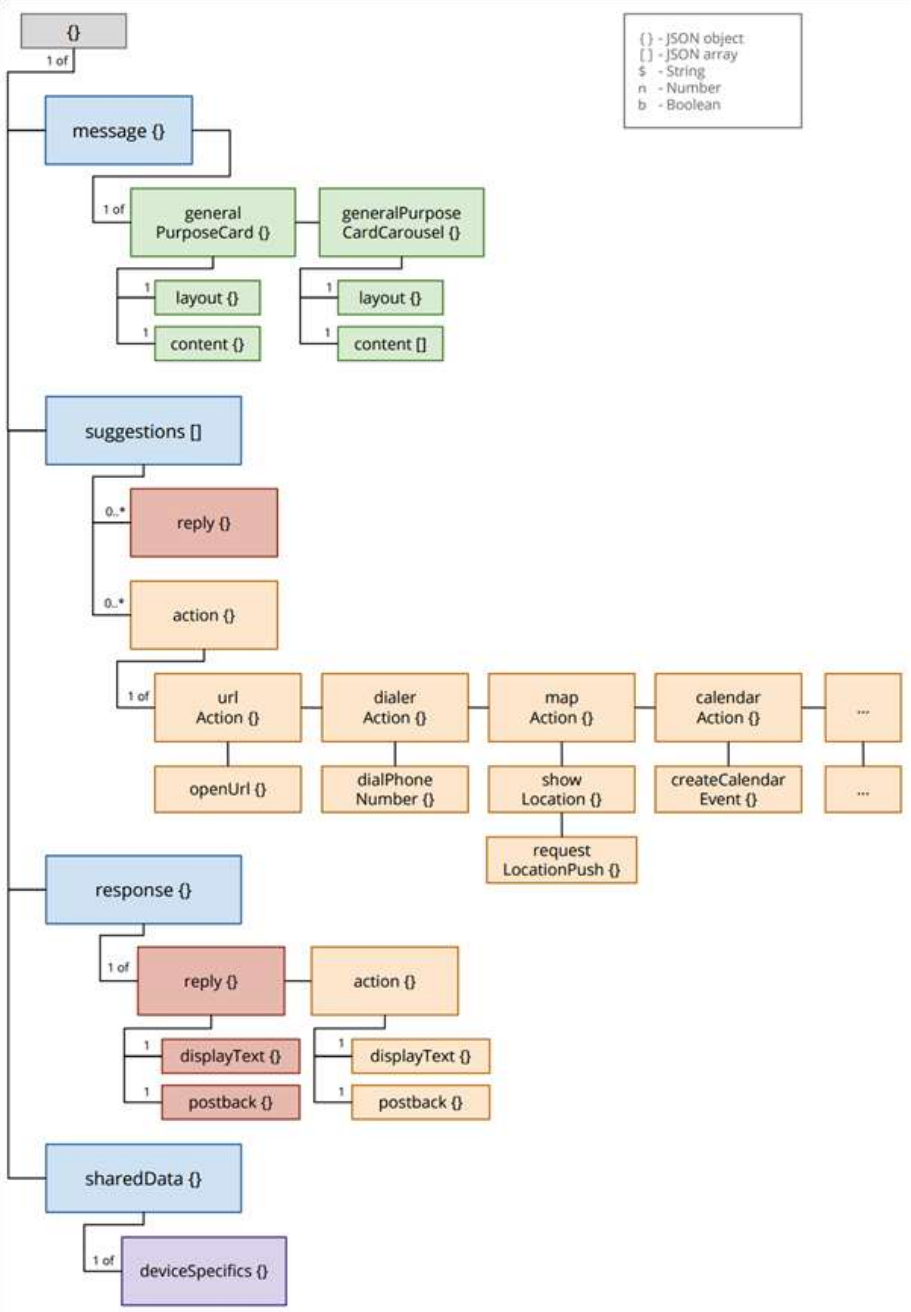



**Figure 14: JSON message payloads data structure overview**

### 3.6.10.4 JSON schema for validation

The following schema defines all JSON payloads exchanged between Chatbot Platform and clients:

```
{
 "$schema": "http://json-schema.org/draft-04/schema#",

 "title": "Root Object",
 "type": "object",
```

```
 "properties": {

  "message": {
    "title": "A chatbot message, sent from chatbot platform to client.",
    "type": "object",
    "oneOf": [{
      "$ref": "#/definitions/messages/generalPurposeCardMessage"
    }, {
      "$ref": "#/definitions/messages/generalPurposeCardCarouselMessage"
    }]
  },

  "suggestions": {
    "title": "Suggested replies and/or suggested actions for a chatbot message, send from chatbot platform to client.",
    "type": "array",
    "items": {
      "oneOf": [{
        "$ref": "#/definitions/suggestions/replies/reply"
      }, {
        "$ref": "#/definitions/suggestions/actions/action"
      }]
    },
    "minItems": 1,
    "maxItems": 11,
    "additionalItems": false
  },

  "response": {
    "title": "Response to a suggested reply or suggested action, sent from client to chatbot platform.",
    "type": "object",
    "oneOf": [{
        "properties": {
          "reply": {
            "$ref": "#/definitions/suggestions/suggestion"
          }
        },
        "required": ["reply"]
      },
      {
        "properties": {
          "action": {
            "$ref": "#/definitions/suggestions/suggestion"
          }
        },
        "required": ["action"]
      }
    ]
  },

  "sharedData": {
    "title": "Data shared by the client with the chatbot platform (e. g. device specifics).",
    "type": "object",
    "oneOf": [{
      "properties": {
        "deviceSpecifics": {
          "$ref": "#/definitions/sharedData/deviceSpecifics"
```

```
        }
      }
    }]
  }

},
"oneOf": [{
    "required": ["message"]
  },
  {
    "required": ["suggestions"]
  },
  {
    "required": ["response"]
  },
  {
    "required": ["sharedData"]
  }
],

"definitions": {

  "messageFragments": {
    "cardMedia": {
      "type": "object",
      "properties": {
        "mediaUrl": {
          "type": "string",
          "format": "uri"
        },
        "mediaContentType": {
          "type": "string"
        },
        "mediaFileSize": {
          "title": "Media file size in bytes",
          "type": "integer",
          "minimum": 0
        },
        "thumbnailUrl": {
          "type": "string",
          "format": "uri"
        },
        "thumbnailContentType": {
          "type": "string"
        },
        "thumbnailFileSize": {
          "title": "Thumbnail file size in bytes",
          "type": "integer",
          "minimum": 0
        },
        "height": {
          "type": "string",
          "enum": ["SHORT_HEIGHT", "MEDIUM_HEIGHT", "TALL_HEIGHT"]
        },
        "contentDescription": {
          "title": "Optional textual description of media content",
          "description": "Accessiblity text for use with screen readers. Will not be shown on screen.",
          "type": "string",
```

```
        "minLength": 1,
        "maxLength": 200
       }
      },
      "required": ["mediaUrl", "mediaContentType", "mediaFileSize", "height"],
      "dependencies": {
       "thumbnailUrl": ["thumbnailContentType", "thumbnailFileSize"]
      }
    },
    "cardTitle": {
     "type": "string",
     "minLength": 1,
     "maxLength": 200
    },
    "cardDescription": {
     "type": "string",
     "minLength": 1,
     "maxLength": 2000
    }
  },

  "messages": {
    "generalPurposeCardMessage": {
     "title": "This defines a general-purpose, standalone Rich Card message.",
     "type": "object",
     "properties": {
       "generalPurposeCard": {
        "type": "object",
        "properties": {
          "layout": {
            "type": "object",
            "oneOf": [{
                "properties": {
                 "cardOrientation": {
                   "type": "string",
                   "enum": ["VERTICAL"]
                 }
                },
                "required": ["cardOrientation"]
              },
              {
                "properties": {
                 "cardOrientation": {
                   "type": "string",
                   "enum": ["HORIZONTAL"]
                 },
                 "imageAlignment": {
                   "type": "string",
                   "enum": ["LEFT", "RIGHT"]
                 }
                },
                "required": ["cardOrientation", "imageAlignment"]
              }
            ]
          },
          "content": {
            "type": "object",
            "properties": {
```

```
            "media": {
              "$ref": "#/definitions/messageFragments/cardMedia"
            },
            "title": {
              "$ref": "#/definitions/messageFragments/cardTitle"
            },
            "description": {
              "$ref": "#/definitions/messageFragments/cardDescription"
            },
            "suggestions": {
              "type": "array",
              "items": {
                "oneOf": [{
                  "$ref": "#/definitions/suggestions/replies/reply"
                }, {
                  "$ref": "#/definitions/suggestions/actions/action"
                }]
              },
              "minItems": 1,
              "maxItems": 4,
              "additionalItems": false
            }
          },
          "anyOf": [{
              "required": ["media"]
            },
            {
              "required": ["title"]
            },
            {
              "required": ["description"]
            }
          ]
        }
      },
      "required": ["layout", "content"]
    }
  },
  "required": ["generalPurposeCard"]
},
"generalPurposeCardCarouselMessage": {
  "title": "This defines a message containing a carousel of general-purpose Rich Cards.",
  "type": "object",
  "properties": {
    "generalPurposeCardCarousel": {
      "type": "object",
      "properties": {
        "layout": {
          "type": "object",
          "properties": {
            "cardWidth": {
              "type": "string",
              "enum": ["SMALL_WIDTH", "MEDIUM_WIDTH"],
              "default": "SMALL_WIDTH"
            }
          },
          "required": ["cardWidth"]
        },
```

```
          "content": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "media": {
                  "$ref": "#/definitions/messageFragments/cardMedia"
                },
                "title": {
                  "$ref": "#/definitions/messageFragments/cardTitle"
                },
                "description": {
                  "$ref": "#/definitions/messageFragments/cardDescription"
                },
                "suggestions": {
                  "type": "array",
                  "items": {
                    "oneOf": [{
                      "$ref": "#/definitions/suggestions/replies/reply"
                    }, {
                      "$ref": "#/definitions/suggestions/actions/action"
                    }]
                  },
                  "minItems": 1,
                  "maxItems": 4,
                  "additionalItems": false
                }
              },
              "anyOf": [{
                  "required": ["media"]
                },
                {
                  "required": ["title"]
                },
                {
                  "required": ["description"]
                }
              ]
            },
            "minItems": 2,
            "maxItems": 10,
            "additionalItems": false
          }
        },
        "required": ["layout", "content"]
      }
    },
    "required": ["generalPurposeCardCarousel"]
  }
},

"suggestionFragments": {
  "postback": {
    "title": "Definition of data to be posted back from UE to chatbot.",
    "type": "object",
    "properties": {
      "data": {
        "type": "string",
```

```
        "maxLength": 2048
      }
    },
    "required": ["data"]
  }
 },

 "suggestions": {
  "suggestion": {
   "title": "Common base definition for suggested replies and suggested actions.",
   "type": "object",
   "properties": {
    "displayText": {
     "type": "string",
     "minLength": 1,
     "maxLength": 25
    },
    "postback": {
     "$ref": "#/definitions/suggestionFragments/postback"
    }
   },
   "required": ["displayText"]
  },
  "replies": {
   "reply": {
    "title": "Definition of a suggested reply.",
    "type": "object",
    "properties": {
     "reply": {
      "allOf": [{
       "$ref": "#/definitions/suggestions/suggestion"
      }]
     }
    },
    "required": ["reply"]
   }
  },
  "actions": {
   "action": {
    "title": "Common base definition of a suggested action.",
    "type": "object",
    "properties": {
     "action": {
      "type": "object",
      "allOf": [{
       "$ref": "#/definitions/suggestions/suggestion"
      }, {
       "oneOf": [{
        "$ref": "#/definitions/suggestions/actions/urlAction"
       }, {
        "$ref": "#/definitions/suggestions/actions/dialerAction"
       }, {
        "$ref": "#/definitions/suggestions/actions/mapAction"
       }, {
        "$ref": "#/definitions/suggestions/actions/calendarAction"
       }, {
        "$ref": "#/definitions/suggestions/actions/composeAction"
       }, {
```

```
            "$ref": "#/definitions/suggestions/actions/deviceAction"
           }, {
            "$ref": "#/definitions/suggestions/actions/settingsAction"
           }]
          }]
         }
        },
        "required": ["action"]
       },
       "urlAction": {
        "title": "Suggested actions to interact a website or deep app link.",
        "properties": {
         "urlAction": {
          "type": "object",
          "oneOf": [{
           "properties": {
            "openUrl": {
             "type": "object",
             "properties": {
              "url": {
               "type": "string",
               "format": "uri"
              }
             },
             "required": ["url"]
            }
           },
           "required": ["openUrl"]
          }]
         }
        },
        "required": ["urlAction"]
       },
       "dialerAction": {
        "title": "Suggested actions for interacting with a phone number.",
        "properties": {
         "dialerAction": {
          "type": "object",
          "oneOf": [{
           "properties": {
            "dialPhoneNumber": {
             "type": "object",
             "properties": {
              "phoneNumber": {
               "type": "string"
              },
              "fallbackUrl": {
               "type": "string",
               "format": "uri"
              }
             },
             "required": ["phoneNumber"]
            }
           },
           "required": ["dialPhoneNumber"]
          }, {
           "properties": {
            "dialEnrichedCall": {
```

```
            "type": "object",
            "properties": {
              "phoneNumber": {
                "type": "string"
              },
              "subject": {
                "type": "string",
                "maxLength": 60
              },
              "fallbackUrl": {
                "type": "string",
                "format": "uri"
              }
            },
            "required": ["phoneNumber"]
          }
        },
        "required": ["dialEnrichedCall"]
      }, {
        "properties": {
          "dialVideoCall": {
            "type": "object",
            "properties": {
              "phoneNumber": {
                "type": "string"
              },
              "fallbackUrl": {
                "type": "string",
                "format": "uri"
              }
            },
            "required": ["phoneNumber"]
          }
        },
        "required": ["dialVideoCall"]
      }]
    }
  },
  "required": ["dialerAction"]
},
"mapAction": {
  "title": "Suggested actions for interacting with a location on a map.",
  "properties": {
    "mapAction": {
      "type": "object",
      "oneOf": [{
        "properties": {
          "showLocation": {
            "title": "Shows a given location on a map.",
            "type": "object",
            "properties": {
              "location": {
                "type": "object",
                "properties": {
                  "latitude": {
                    "type": "number"
                  },
                  "longitude": {
```

```
                    "type": "number"
                  },
                  "label": {
                    "type": "string",
                    "minLength": 1,
                    "maxLength": 100
                  },
                  "query": {
                    "title": "Search for location(s) by query",
                     "description": "Search is based on user's current location",
                     "examples": [
                       "restaurants",
                       "GSMA Head Office, 25 Walbrook, London, UK"
                     ],
                    "type": "string",
                    "minLength": 1,
                    "maxLength": 200
                  }
                },
                "oneOf": [{
                    "required": ["latitude", "longitude"]
                  },
                  {
                    "required": ["query"]
                  }
                ]
              },
              "fallbackUrl": {
                "type": "string",
                "format": "uri"
              }
            },
            "required": ["location"]
          }
        },
        "required": ["showLocation"]
      },
      {
        "properties": {
          "requestLocationPush": {
            "title": "One-time request to send a geo location push from UE to chatbot",
            "type": "object"
          }
        },
        "required": ["requestLocationPush"]
      }
    ]
  }
 },
 "required": ["mapAction"]
},
"calendarAction": {
 "title": "Suggested actions for interacting with a calendar event.",
 "properties": {
  "calendarAction": {
   "type": "object",
   "oneOf": [{
    "properties": {
```

```
            "createCalendarEvent": {
              "type": "object",
              "properties": {
                "startTime": {
                  "type": "string",
                  "format": "date-time"
                },
                "endTime": {
                  "type": "string",
                  "format": "date-time"
                },
                "title": {
                  "type": "string",
                  "minLength": 1,
                  "maxLength": 100
                },
                "description": {
                  "type": "string",
                  "minLength": 1,
                  "maxLength": 500
                },
                "fallbackUrl": {
                  "type": "string",
                  "format": "uri"
                }
              },
              "required": ["startTime", "endTime", "title"]
            }
          },
          "required": ["createCalendarEvent"]
        }]
      }
    },
    "required": ["calendarAction"]
  },
  "composeAction": {
    "title": "Suggested actions for composing draft messages.",
    "properties": {
      "composeAction": {
        "type": "object",
        "oneOf": [{
          "properties": {
            "composeTextMessage": {
              "title": "Compose a draft text message.",
              "type": "object",
              "properties": {
                "phoneNumber": {
                  "type": "string"
                },
                "text": {
                  "type": "string",
                  "maxLength": 100
                }
              },
              "required": ["phoneNumber", "text"]
            }
          },
          "required": ["composeTextMessage"]
```

```
          },
          {
            "properties": {
              "composeRecordingMessage": {
                "title": "Compose a draft message with a media recording.",
                "type": "object",
                "properties": {
                  "phoneNumber": {
                    "type": "string"
                  },
                  "type": {
                    "type": "string",
                    "enum": ["AUDIO", "VIDEO"]
                  }
                },
                 "required": ["phoneNumber", "type"]
              }
            },
            "required": ["composeRecordingMessage"]
          }
        ]
      }
    },
    "required": ["composeAction"]
  },
  "deviceAction": {
    "title": "Suggested actions for interacting with the user's device.",
    "properties": {
      "deviceAction": {
        "type": "object",
        "oneOf": [{
          "properties": {
            "requestDeviceSpecifics": {
              "title": "Request specifics about the user's device.",
              "type": "object"
            }
          },
          "required": ["requestDeviceSpecifics"]
        }
       ]
      }
    },
    "required": ["deviceAction"]
  },
  "settingsAction": {
    "title": "Suggested actions for interacting with app settings",
    "properties": {
      "settingsAction": {
        "type": "object",
        "oneOf": [{
          "properties": {
            "disableAnonymization": {
              "title": "Ask the user to disable the anonymization setting.",
              "type": "object"
            }
          },
          "required": ["disableAnonymization"]
        },
```

```
            {
              "properties": {
                "enableDisplayedNotifications": {
                  "title": "Ask the user to enable sending displayed notifications.",
                  "type": "object"
                }
              },
              "required": ["enableDisplayedNotifications"]
            }
          ]
        }
      },
      "required": ["settingsAction"]
    }
  }
 },

 "sharedData": {
  "deviceSpecifics": {
    "title": "Device specifics shared by the client with the chatbot platform.",
    "type": "object",
    "properties": {
      "deviceModel": {
        "title": "Short description of the device model.",
        "type": "string",
        "minLength": 1,
        "maxLength": 10
      },
      "platformVersion": {
        "title": "Version information about the operating system on the device .",
        "type": "string",
        "minLength": 1,
        "maxLength": 20
      },
      "clientVendor": {
        "title": "Short code for client vendor, same as used during RCS autoconfiguration.",
        "type": "string",
        "minLength": 1,
        "maxLength": 4
      },
      "clientVersion": {
        "title": "Version information about the client, same as used during RCS autoconfiguration ",
        "type": "string",
        "minLength": 1,
        "maxLength": 15
      },
      "batteryRemainingMinutes": {
        "title": "Remaining battery use of device in minutes",
        "type": "integer",
        "minimum": 0
      }
    }
  }
 }
}
```

**Table 67: JSON message payloads schema**

#### 3.6.10.5 Types of Rich Cards

This specification supports two different kinds of Chatbot Messages which are sent from Chatbot Platforms to clients:

- Single Rich Card
- Carousel of Rich Cards

NOTE: Chatbots can also send regular RCS messages as stated in section 3.6.10.1.

The following sections describe the two different kinds of Chatbot Messages.

##### 3.6.10.5.1 Single Rich Card

###### 3.6.10.5.1.1 JSON data format diagram

Note: This overview doesn't contain all fields and possible sub-objects. See the JSON schema in section 3.6.10.4 for more details.

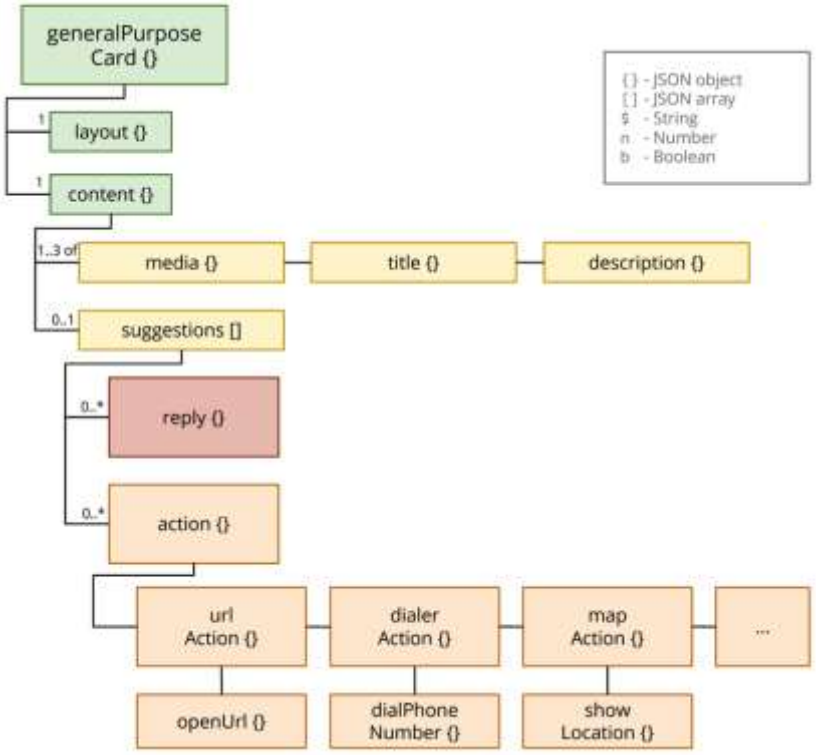



**Figure 15: JSON Single Rich Card data structure overview**

###### 3.6.10.5.1.2 Example JSON payload

The following payload is an example for a Single Rich Card including suggested replies and suggested actions:

```
{
 "message": {
  "generalPurposeCard": {
   "layout": {
    "cardOrientation": "HORIZONTAL",
    "imageAlignment": "LEFT"
   },
   "content": {
```

```
      "media": {
        "mediaUrl": "https://cdn.server/path/media.mp4",
        "mediaContentType": "video/mp4",
        "mediaFileSize": 2718288,
        "thumbnailUrl": "https://cdn.server/path/media.png",
        "thumbnailContentType": "image/png",
        "thumbnailFileSize": 314159,
        "height": "MEDIUM_HEIGHT",
        "contentDescription": "Textual description of media content, e. g. for use with screen readers."
      },
      "title": "This is a single rich card.",
      "description": "This is the description of the rich card. It's the first field that will be truncated if it exceeds the maximum width or height of a card.",
      "suggestions": [{
          "reply": {
            "displayText": "No",
            "postback": {
              "data": "set_by_chatbot_reply_no"
            }
          }
        },
        {
          "action": {
            "urlAction": {
              "openUrl": {
                "url": "https://www.google.com"
              }
            },
            "displayText": "Open website or deep link",
            "postback": {
              "data": "set_by_chatbot_open_url"
            }
          }
        },
        {
          "action": {
            "dialerAction": {
              "dialPhoneNumber": {
                "phoneNumber": "+1650253000"
              }
            },
            "displayText": "Call a phone number",
            "postback": {
              "data": "set_by_chatbot_open_dialer"
            }
          }
        }
      ]
    }
  }
 }
}
```


**Table 68: Chatbot communication Single Rich Card example**

#### 3.6.10.5.2 Carousel of Rich Cards

##### 3.6.10.5.2.1 JSON data format diagram

NOTE: This overview doesn't contain all fields and possible sub-objects. See the JSON schema in section 3.6.10.4 above for more details.

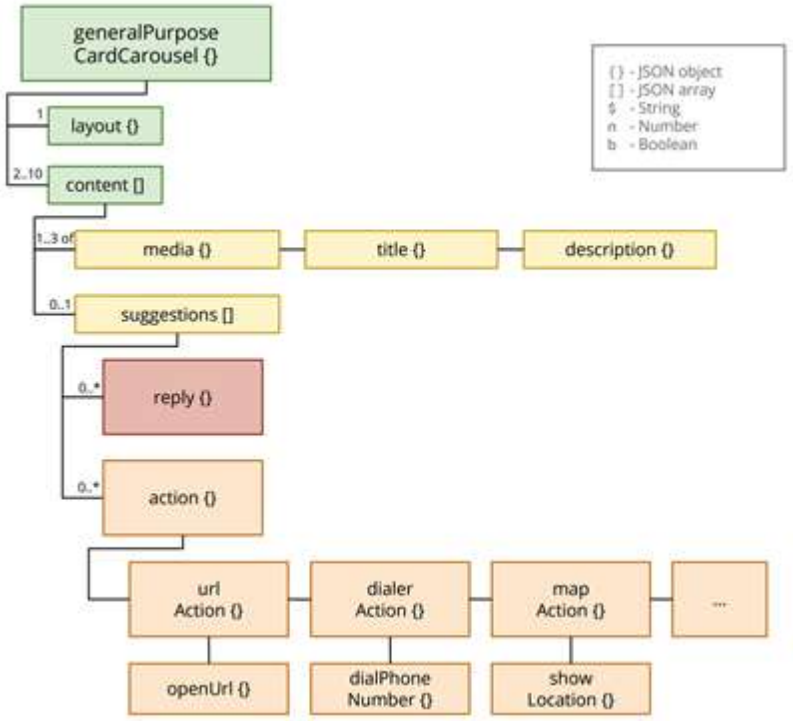



**Figure 16: JSON Carousel of Rich Cards data structure overview**

##### 3.6.10.5.2.2 Example JSON payload

Two or more (up to ten) cards can be combined into a carousel. The following payload is an example for a carousel of cards including suggested replies and suggested actions:

```
{
 "message": {
  "generalPurposeCardCarousel": {
   "layout": {
    "cardWidth": "MEDIUM_WIDTH"
   },
   "content": [{
     "media": {
      "mediaUrl": "https://cdn.server/path/media.mp4",
      "mediaContentType": "video/mp4",
      "mediaFileSize": 2718288,
      "thumbnailUrl": "https://cdn.server/path/media.png",
      "thumbnailContentType": "image/png",
      "thumbnailFileSize": 314159,
      "height": "SHORT_HEIGHT",
      "contentDescription": "Textual description of media content, e. g. for use with screen readers."
     },
     "title": "This is the first rich card in a carousel.",
     "description": "This is the description of the rich card. It's the first field that will be truncated if it exceeds the maximum width or height of a card.",
     "suggestions": [{
```

```
          "action": {
            "mapAction": {
              "showLocation": {
                "location": {
                  "latitude": 37.4220041,
                  "longitude": -122.0862515,
                  "label": "Googleplex"
                },
                "fallbackUrl": "https://www.google.com/maps/@37.4219162,-122.078063,15z"
              }
            },
            "displayText": "Show location on a map",
            "postback": {
              "data": "set_by_chatbot_open_map"
            }
          }
        },
        {
          "action": {
            "calendarAction": {
              "createCalendarEvent": {
                "startTime": "2017-03-14T00:00:00Z",
                "endTime": "2017-03-14T23:59:59Z",
                "title": "Meeting",
                "description": "GSG review meeting"
              }
            },
            "displayText": "Schedule Meeting",
            "postback": {
              "data": "set_by_chatbot_create_calendar_event"
            }
          }
        }
      ]
    },
    {
      "title": "This is the second rich card in the carousel.",
      "description": "Carousel cards need to specify a card width in the 'layout' section. For small width cards, only short and medium height media are supported.",
      "[...]": "[...]"
    }
  ]
 }
}
}
```


**Table 69: Chatbot Communication Carousel Rich card Example**

#### 3.6.10.5.3 Client Processing of Card Media of Rich Cards

If the client receives a Chatbot message containing Media objects, then the client shall download the files identified via the values of the mediaUrl and thumbnailUrl properties of a Media object via the file download procedure defined in section 3.2.5.3.2.1.

### 3.6.10.6 Suggested Chip List

As specified in section 3.6.10.1, a Chatbot Platform shall provide a Suggested Chip List always in combination with the message it is associated to.

#### 3.6.10.6.1 Payload from Chatbot Platform to Clients

##### 3.6.10.6.1.1 Suggested Replies

Suggested Replies consist of a display text and a set of postback data.

##### 3.6.10.6.1.2 Suggested Actions

Suggested Actions are grouped into seven different categories supporting a total of twelve different suggested actions:

- urlAction
  - openUrl — opens a web site or app via deep linking
- dialerAction
  - dialPhoneNumber — calls a phone number via the user's dialler app
  - dialEnrichedCall — start an Enriched Call via the user’s dialler app
  - dialVideoCall — start a video call via the user’s dialler app
- mapAction
  - showLocation — show location(s) on a map for given coordinates or search query
  - requestLocationPush — request for a one-time geo location push
- calendarAction
  - createCalendarEvent — creates a new event on the user's calendar
- composeAction
  - composeTextMessage — compose a draft text message
  - composeRecordingMessage — compose a draft message and start recording audio or video
- deviceAction
  - requestDeviceSpecifics — request for a one-time share of device specifics (device model, operating system version, messaging client identifier and version, and remaining battery charge in minutes)
- settingsAction
  - disableAnonymization — ask the user to disable the anonymization setting
  - enableDisplayedNotifications — ask the user to enable sending displayed notifications

This design allows for easily extending action categories and concrete actions in the future.

Most actions allow fallback URLs in case a user does not have any app of the required type installed. Chatbot platforms can use the fallback URL to suggest an appropriate app to the user.

#### 3.6.10.6.1.3 Example JSON payload

The following payload defines a Suggested Chip List with two suggested replies and all currently supported actions:

```
{
 "suggestions": [{
    "reply": {
     "displayText": "Yes",
     "postback": {
       "data": "set_by_chatbot_reply_yes"
     }
    }
  },
  {
    "reply": {
     "displayText": "No",
     "postback": {
       "data": "set_by_chatbot_reply_no"
     }
    }
  },
  {
    "action": {
     "urlAction": {
       "openUrl": {
         "url": "https://www.google.com"
       }
     },
     "displayText": "Open website or deep link",
     "postback": {
       "data": "set_by_chatbot_open_url"
     }
    }
  },
  {
    "action": {
     "dialerAction": {
       "dialPhoneNumber": {
         "phoneNumber": "+1650253000"
       }
     },
     "displayText": "Call a phone number",
     "postback": {
       "data": "set_by_chatbot_dial_phone_number"
     }
    }
  },
  {
    "action": {
     "dialerAction": {
       "dialEnrichedCall": {
         "phoneNumber": "+1650253000",
         "subject": "The optional subject for the enriched call"
       }
     },
     "displayText": "Start enriched call",
     "postback": {
       "data": "set_by_chatbot_dial_enriched_call"
```

```
      }
    }
  },
  {
    "action": {
      "dialerAction": {
        "dialVideoCall": {
          "phoneNumber": "+1650253000"
        }
      },
      "displayText": "Start video call",
      "postback": {
        "data": "set_by_chatbot_dial_video_call"
      }
    }
  },
  {
    "action": {
      "mapAction": {
        "showLocation": {
          "location": {
            "latitude": 37.4220041,
            "longitude": -122.0862515,
            "label": "Googleplex"
          },
          "fallbackUrl": "https://www.google.com/maps/@37.4219162,-122.078063,15z"
        }
      },
      "displayText": "Show location on a map",
      "postback": {
        "data": "set_by_chatbot_show_location"
      }
    }
  },
  {
    "action": {
      "mapAction": {
        "showLocation": {
          "location": {
            "query": "restaurants"
          },
          "fallbackUrl": "https://www.google.com/maps/search/restaurants"
        }
      },
      "displayText": "Search location(s) on map",
      "postback": {
        "data": "set_by_chatbot_search_locations"
      }
    }
  },
  {
    "action": {
      "mapAction": {
        "requestLocationPush": {}
      },
      "displayText": "Request a geo location",
      "postback": {
        "data": "set_by_chatbot_request_location_push"
```

```
      }
    }
  },
  {
    "action": {
      "calendarAction": {
        "createCalendarEvent": {
          "startTime": "2017-03-14T00:00:00Z",
          "endTime": "2017-03-14T23:59:59Z",
          "title": "Meeting",
          "description": "GSG review meeting"
        }
      },
      "displayText": "Schedule Meeting",
      "postback": {
        "data": "set_by_chatbot_create_calendar_event"
      }
    }
  },
  {
    "action": {
      "composeAction": {
        "composeTextMessage": {
          "phoneNumber": "+1650253000",
          "text": "Draft to go into the send message text field."
        }
      },
      "displayText": "Draft a text message",
      "postback": {
        "data": "set_by_chatbot_compose_text_message"
      }
    }
  },
  {
    "action": {
      "composeAction": {
        "composeRecordingMessage": {
          "phoneNumber": "+1650253000",
          "type": "VIDEO"
        }
      },
      "displayText": "Record audio or video",
      "postback": {
        "data": "set_by_chatbot_compose_recording_message"
      }
    }
  },
  {
    "action": {
      "deviceAction": {
        "requestDeviceSpecifics": {}
      },
      "displayText": "Request device specifics",
      "postback": {
        "data": "set_by_chatbot_request_device_specifics"
      }
    }
  },
```

```
  {
    "action": {
      "settingsAction": {
        "disableAnonymization": {}
      },
      "displayText": "Share your phone number",
      "postback": {
        "data": "set_by_chatbot_disable_anonymization"
      }
    }
  },
  {
    "action": {
      "settingsAction": {
        "enableDisplayedNotifications": {}
      },
      "displayText": "Send read receipts",
      "postback": {
        "data": "set_by_chatbot_enable_displayed_notifications"
      }
    }
  }
 ]
}
```

**Table 70: Chatbot Communication Suggested Chip List example**

#### 3.6.10.6.2 Payload from Client to Chatbot Platform

Whenever a user selects a suggested reply or suggested action, the RCS client shall send a special response message back to the Chatbot with the following precisions:

1. The message shall be a CPIM message.
2. The content type of the CPIM body of this message shall be set to the value defined in section 3.6.10.2.3.
3. The CPIM body of the response message shall be set as

    a) defined in section 3.6.10.6.2.1 for suggested replies responses
    b) as defined in section 3.6.10.6.2.2 for suggested actions responses.

For the `requestDeviceSpecifics` suggested action, RCS clients share data with Chatbot Platforms in a separate payload. The behaviour is defined as follows:

1. The Chatbot sends a `requestDeviceSpecifics` suggested action to the RCS client.
2. The user taps selects the suggested action.
3. The RCS client sends a response to the suggested action including the postback data (see section 3.6.10.6.2.2 below).
4. The user optionally interacts with a client-defined user interface to confirm sharing data with the Chatbot.
5. The RCS client sends the shared data in a separate payload (see section 3.6.10.6.2.3 below).

##### 3.6.10.6.2.1 Response for suggested replies

When sending a response message pertaining to a user interaction with a suggested reply, an RCS client shall generate the body of the response as follows:

1. Generate a `response` object
2. Include in this `response` object, a `reply` object including all properties that were provided in the suggested reply (such as the `postback` object).

The following payload is an example of a response for a suggested reply from a client to a Chatbot Platform:

```
{
"response": {
  "reply": {
   "displayText": "No",
   "postback": {
    "data": "set_by_chatbot_reply_no"
   }
  }
 }
}
```

**Table 71: Chatbot Communication Suggested Reply example**

The RCS client shall present to the user the value of the `displayText` property of the suggested reply as a message that was sent.

#### 3.6.10.6.2.2 Response for suggested actions

Following a user interaction with a suggested action, an RCS client shall send a response to the Chatbot Platform if the suggested action that was used included a postback data object. In this case, the RCS client shall generate the body of the response as follows:

1. Generate a `response` object
2. Include in this `response` object, the `displayText` property and the full `postback` object that were provided for this suggested action.

The following payload is an example of a response for a suggested action from a client to a Chatbot Platform:

```
{
 "response": {
  "action": {
   "displayText": "Open website or deep link",
   "postback": {
    "data": "set_by_chatbot_open_url"
   }
  }
 }
}
```

**Table 72: Chatbot Communication Suggested Action example**

The RCS client shall not present this message to the user as a message that was sent.

#### 3.6.10.6.2.3 Shared client data

When the user agrees to share client data following a `requestDeviceSpecifics` suggested action, the client shall send a `sharedData` payload with a `deviceSpecifics` property as a CPIM Message to the Chatbot Platform. In this payload, the client shall set

- The value of the `deviceModel` property to the same value as provided for the `terminal_model` Configuration request parameter defined in section 2.4 of [PRD-RCC.14]

- The value of the `platformVersion` property to the same value as provided for the `terminal_sw_version` defined in section 2.4 of [PRD-RCC.14]
- The value of the `clientVendor` property to the same value as provided for the `client_vendor` Configuration request parameter defined in section 2.3.2.2
- The value of the `clientVersion` property to the same value as provided for the `client_version` Configuration request parameter defined in section 2.3.2.2
- The value of the `batteryRemainingMinutes` property to the estimated number of remaining minutes of battery life

The Content-Type of the CPIM body shall be set as specified in section 3.6.10.2.4.

The following payload is an example of data that clients share with a Chatbot Platform as a result of the user interacting with the `requestDeviceSpecifics` suggested action:

```
{
 "sharedData": {
  "deviceSpecifics": {
   "deviceModel": "SmartPhone",
   "platformVersion": "Android-7.1.2-N481G3",
   "clientVendor": "VNDR",
   "clientVersion": "RCSAndrd-1.0",
   "batteryRemainingMinutes": 517
  }
 }
}
```

**Table 73: Chatbot Communication Shared Client data example**

### 3.6.11 Critical Chatbots

The list of critical Chatbots shall be defined by the Service Provider and stored in the list of Chatbots requiring specific management in the format defined in section 3.6.3.3.

The list of critical Chatbots shall be identified as CRITICAL (see <name of the list> in section 3.6.3.3). The specific marker to identify the list shall therefore be:

```
LIST:CRITICAL<CRLF>
<CRLF>
```

When the list of Chatbots requiring specific management includes a list of critical Chatbots via the procedures defined in section 3.6.3.3, the client shall store the list of critical Chatbots locally.

When receiving a 1-to-1 Chatbot session request as per section 3.6.8.4, from a Chatbot whose URI is in the critical Chatbots List, the client shall accept the session.

When initiating a 1-to-1 Chatbot session request as per section 3.6.8.3, with a Chatbot whose URI is in the critical Chatbots List, the client shall never request privacy as per section 3.6.5.1.2.1.

The URI comparisons shall be done in the client without considering the 'tk' parameter.

# 4 Cross-service functionality

## 4.1 Common Message Store

### 4.1.1 Overview

RCS supports a “Common Message Store” as described in Section 5.5 of [CPM-SYS_DESC] and specified in [CPM-MSGSTOR-REST]. Using an HTTP RESTful interface, an RCS client can access and manage stored objects, as described in [CPM-MSGSTOR-REST] regardless of their RCS service registration.

The Restful resources from Client to Message Store Server are executed either during an “initial sync” i.e. First synchronization between client and store, or following a notification to the client of store content changes. The Restful design uses a notification channel as its main method of understanding when to synchronize, and this greatly helps with battery life on the client (there are few connection requests to the Message Store Server and less need for periodic synchronising).

Regarding the client synchronisation mechanism that applies, client synchronization guidelines are described in section 4.1.16.7.

A 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 synchronisation 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.

Sections 4.1.8, 4.1.9, 4.1.10 and 4.1.11 describe the mechanism used for a device to correlate legacy SMS/MMS messages with the same messages already stored in the Common Message Store.

The Common Message Store shall support storage and retrieval of RCS and xMS message objects and call log objects as defined in [CPM-MSGSTOR-REST].

### 4.1.2 Support of GBA in the Common Message Store

The General Bootstrapping Architecture (GBA) defined in [3GPP TS 33.220] provides mechanisms for AKA based user authentication using the 3GPP Authentication Centre (AuC) and the USIM or ISIM. The Common Message Store supports the authentication of primary devices via GBA with the extension defined in this section.

The use of the GBA is based on an existing bootstrapped security association managed between the client and the Bootstrapped Security Function (BSF) operated by the Service Provider. The bootstrapped security association provides the client with a Bootstrapping Transaction Identifier (B-TID) and key material which can be used to authenticate the user with the Service Provider's network.

The procedures for the client and the Common Message Store shall follow the definitions of [3GPP TS 24.109].

### 4.1.3 Support of OpenID Connect in the Common Message Store

If the client receives in result of an API request to the Common Message Store a HTTP 302 FOUND response, the client shall change the type of the request to HTTP GET and follow the procedures for OpenID Connect based authentication as defined in section 2.12.2. The procedure results in a reconnection back to the Common Message Store commencing in the processing of the API request.

If the client receives a 403 FORBIDDEN response in the result of the processing, then the client is not authorised to use the Common Message Store.

### 4.1.4 Support for Digest Authentication

If the value of the MESSAGE STORE AUTH client configuration parameter is set to 2 and the client receives in result of an HTTP request for the RESTful interface of the Common Message Store a HTTP 401 AUTHENTICATION REQUIRED error response carrying a WWW Authenticate header indicating the Digest scheme without the 3GPP-bootstrapping indication as defined in [3GPP TS 24.109], then the client shall invoke the Digest authentication procedures as per [RFC2716] sending a subsequent request with the Authorization header. An RCS client shall include the qop directive if provided by the server.

The client shall compose the Authorization header using the value derived from the MESSAGE STORE USERNAME client configuration parameter defined in section A.1.3 for the username and the value derived from the MESSAGE STORE PASSWORD client configuration parameter defined in section A.1.3 for the password.

### 4.1.5 Support for Basic Authentication

If the value of the MESSAGE STORE AUTH parameter is set to 1, then the client shall, in accordance with the definition of [RFC2716], pre-emptively send a HTTP Authorization header in all HTTP requests using a secure connection and targeting the protection space identified by the URL derived from configuration parameter MESSAGE STORE URL defined in section A.1.3.

If the value of the MESSAGE STORE AUTH parameter is set to 2 and the client receives in result of an HTTP request for the RESTful interface of the Common Message Store a HTTP 401 AUTHENTICATION REQUIRED error response carrying a WWW Authenticate header field indicating the Basic scheme, then the client shall invoke the procedures for Basic Authentication as defined in [RFC2716]. If successfully authenticated, the client shall from then on pre-emptively send a HTTP Authorization header in all HTTP requests using a secure connection and targeting the protection space identified by the "realm" provided by the Common Message Store.

The client shall compose the Authorization header using the value derived from MESSAGE STORE USERNAME parameter defined in section A.1.3 for user-id and the value derived from the MESSAGE STORE PASSWORD parameter defined in section A.1.3 for the password.

### 4.1.6 RESTful Web Service Calls

The web service endpoints for the Message Store Server are members of two APIs.

- The Network Message Store (NMS) API defined in [CPM-MSGSTOR-REST] defines endpoints for managing messages, folders and for creating and managing subscriptions.
- The Notification Channel (NC) API defined in [CPM-MSGSTOR-REST] defines endpoints for creating and managing notification channels suitable for use with NMS polling. Using the NC API, you can create long polling channels.

#### 4.1.6.1 Example NMS API Call

Prototype (base API call)

```
https://{nmsHost}/nms/v1/base/{boxId}/{endpoint}
```

Example

```
https://nms-mc4.enc.MNO.net/nms/v1/base/tel+12021001000/objects/6E
```

All elements are case sensitive and are as below:

| Part | Type | Value/Example |
|---|---|---|
| {nmsHost} | Supplied from the RCS configuration document Parameter NMS_URL | https://nmsHost |
| API | Constant | nms |
| API Version | Variable | v1 |
| Store | Constant | base |
| {boxId} | Variable (user identifier) | tel:+12021001000 |
| {endpoint} | Desired endpoint | /objects/{objectId} (endpoint prototype)<br>objects/6E (example) |

#### 4.1.6.2 API Version

Many calls require a version ID to be included in the call's URL path. The prescribed version of the interface as per [CPM-MSGSTOR-REST] is "v1". The examples provided will include this version ID as a static portion of the call's URL. Any attempts to request an unsupported version will result in the server responding with the supported versions available.

#### 4.1.6.3 Box ID

The subscriber number with the country prefix i.e. "tel: E164".

#### 4.1.6.4 A Notification Channel (NC) API Call

Prototype

```
https://{ncHost}/notificationchannel/v1/{userid}/{endpoint}
```

Example

```
https://nms-
mc4.enc.mno.net/notificationchannel/v1/tel:+12021001000/channels/f7d8c62c-
969b-4090-896f-67708f615a89
```

All elements are case sensitive and are as below:

| Part | Type | Value/Example |
|---|---|---|
| {nmsHost} | Supplied from the RCS configuration document parameter NMS_NC_URL | https://ncHost |
| API | Constant | notificationchannel |
| API Version | Variable | v1 |
| Store | Constant | base |
| {userId} | Variable (user identifier) | tel:+12021002003 |
| {endpoint} | Desired endpoint and IDs | channels/f7d8c62c-969b-4090-896f-67708f615a89 |

Version and UserId are as defined for the NMS API.

### 4.1.7 Folder Structure

The message stores messages in folders as follows:

- RCS/CPM and legacy one-to-one messages are stored in folders based on
  - the identity of the contact (for 1-to-1 messages)
  - the Conversation-ID (for group messages, i.e. 1-to-N messages).

Clients should facilitate folder selection by using the Is-CPM-Group parameter (set to “yes”) when depositing RCS group messages. Clients should use “no” for all other messages.

To discover the folder structure, clients should follow the process described in [CPM-MSGSTOR-REST] section 6.1.3 – i.e. descend recursively from root. The /folders/operations/search resource should be used to find the root folder.

#### 4.1.7.1 Folder identifies

As per [CPM-MSGSTOR-REST] section 6.1.3 the root folder of the user’s mailbox is the top level of the folder tree and shall be uniquely identified i.e. it shall have an attribute called “root” with the value “yes”. There is only one root folder, and the name of the root folder is an empty string unless specifically assigned to be some other name by the Message Storage Server.

The folder structure is shown below in Figure 17. The format of the conversation folder names shall be as follows:

- For RCS objects relating to a 1-to-1 conversation (i.e. file transfer objects, session info objects, message objects related to Standalone Messages, 1-to-1 Chat messages, and legacy messages (i.e. SMS/MMS)) shall be stored in a folder identified by the identity of the contact in the conversation. This identity shall be determined from the P-Asserted-identity of the contact, received by the Messaging Server via the incoming SIP request towards the served user, or via the incoming SIP response towards the served user.
  - If a global E.164 phone number is available and there is no 'tk' parameter present, that shall be used in the global-number representation of a tel URI as defined in [RFC3966]. Parameters are not used in the folder name.
    Example: tel:+4917123456789

NOTE: RCS user clients storing messages in the Common Message Store may not have sufficient information to normalize all destination addresses to global E.164 phone number format. In this case the folder name may be formatted as defined below for non E.164 numbers.

- If a non E.164 phone number is available and there is no 'tk' parameter present, it shall be used in the local-number representation of a tel URI as defined in [RFC3966]. These numbers are typically short codes used for addressing of value added services outside the public numbering plan. Parameters and context are not used in the folder name.
  Example: tel:22632677
- Phone numbers are typically derived from
  - tel URI in SIP Signalling
  - SIP URI in SIP signalling if appended by user=phone parameter
  - SMS Originator Address
  - SMS Destination Address
  - MMS Originator Address
  - MMS Recipient Address
- If both a SIP URI and tel URI are present and there is a 'tk' URI parameter, the SIP URI shall be used as defined in [RFC3261] after converting it to lower case first. The 'tk' URI parameter (see section 2.5.4.3) shall be included in the folder name.
  Examples: sip:chatboty@botplatform.botplatformz.com;tk=off, or sip:chatboty@botplatform.botplatformz.com;tk=on
- If only a SIP URI is available the SIP URI shall be used as defined in [RFC3261] after converting it to lower case first. If present the 'tk' URI parameter (see section 2.5.4.3) shall be included in the folder name.
  Examples: sip:joe@example.com, or sip:chatboty@botplatform.botplatformz.com;tk=off, or sip:chatboty@botplatform.botplatformz.com;tk=on
- If an e-mail address is available it shall be used as mailto URI [RFC6068] after converting characters to lower case first.
  Example: mailto:joe@example.com
  e-mail addresses are typically derived from
  - MMS Originator Address
  - MMS Recipient Address
- If an alphanumeric address is available it shall be used with no conversion. These addresses are typically display names from SMS value added services without a routing address.
  Example: ACME Corporation
  Alphanumerical addresses are typically derived from
  - SMS Originator Address

Entities managing folder names in the Common Message Store shall comply with the mailbox international naming conventions defined in [RFC3501].

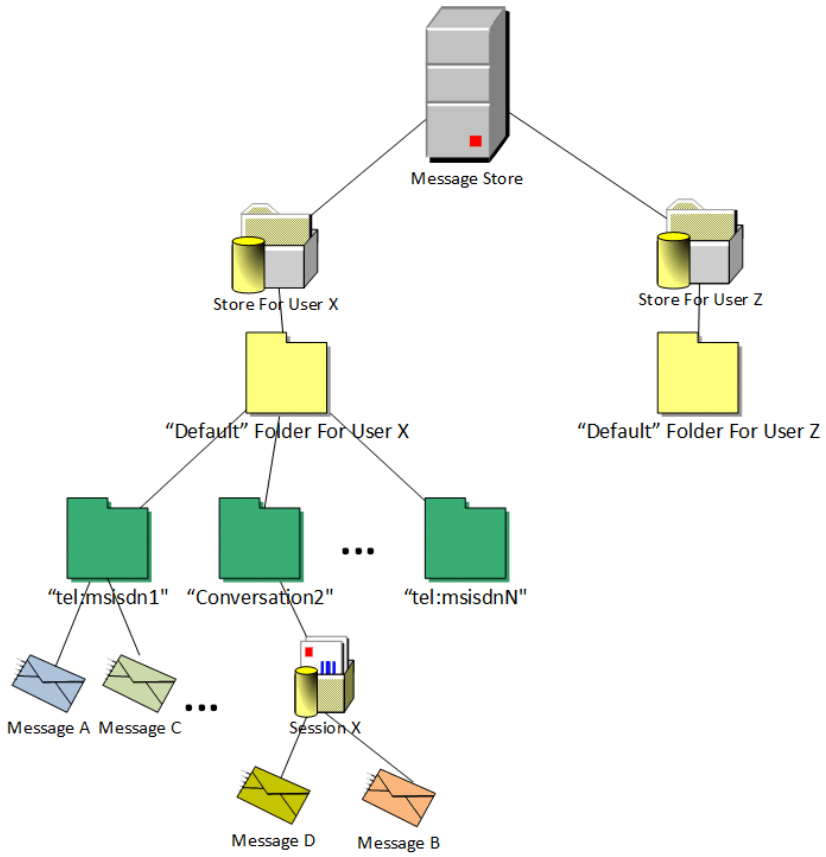



**Figure 17: Example Folder Structure**

#### 4.1.7.2 Storage of IMDN Objects,

IMDNs are deposited to the message storage server with a Message Context of "imdn-message" and synchronized to the clients as individual message objects. The IMDN message object contains an attribute that links it to the original message (DispositionOriginalMessageID), see section 5.4.10 of [CPM-MSGSTOR-REST].

### 4.1.8 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 or stored as 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 or stored as SMS
  - multimedia-message: the message was sent via MMS

#### 4.1.8.1 Client initiated storage of SMS/MMS

The RCS Client stores messages in the Default folder.

NOTE: This procedure is applied when synchronising as described in section 4.1.16.7.

When storing a SMS or MMS message and their delivery and read reports in the Common Message Store the client shall follow the definitions for recording of messages in sections 4.1.12 and 4.1.13 respectively.

If the SMS MESSAGE STORE or MMS MESSAGE STORE configuration parameter defined in Table 85 is set to “always store in the Common Message Store" (i.e. 2) then if storing them locally the client shall upload new sent or received messages and for sent messages, their delivery and read reports (SMS or MMS respectively) to the Common Message Store and link them with the local message. When storing a short message in the Common Message Store the client should not set the correlation information defined in section 4.1.9 for SMS. When storing a multimedia message, a delivery or read report in the Common Message Store the client shall set the correlation information defined in section 4.1.9 for MMS.

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) then the client shall apply the message correlation for new sent or received messages and their delivery and read reports if storing them locally as described in section 4.1.9 for SMS and MMS respectively. If the client determines that a locally stored message and its delivery and read report (for sent SMS or MMS respectively) is not already stored in the Common Message Store, the client shall

- store this message to the Common Message Store with the correlation information defined in section 4.1.9 (for SMS and MMS) respectively, and
- Link it with the local message.

The client shall store SMS and MMS messages and their delivery and read reports in the Common Message Store in a folder identified by the identity of the Contact in the Conversation under the Default folder. If this folder is not yet created in the Default folder, the client shall first create it. For definition of the folder names refer to section 4.1.7.

When storing a message sent or received as an SMS/MMS in the Common Message Store, the client shall set

- a Message-Context header and attribute to pager-message/multimedia-message as described in section 4.1.8 and
- The Message-Direction header and Direction attribute as described in section 4.1.16.8.

### 4.1.9 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 (as defined in section 4.1.10) in the Message-Correlator header of the message (see [CPM-MSGSTOR-REST]).
- The device shall use this Message-Correlator header value, along with To/From headers to find the corresponding locally stored SMS message
- The algorithm is as described in section 4.1.10.

For MMS messages:

- Each MMS message and the corresponding delivery reports are defined by a unique MMS Message ID
- The entity in the network storing the message and corresponding delivery reports shall store the MMS Message ID in the Message-Correlator header defined in [CPM-MSGSTOR-REST]. The header value shall contain the Message-ID-value defined in [MMSENC] as "ascii-value".
- The device uses the MMS Message ID from MMS messages and delivery reports to find the Unique Identifier (UID) of corresponding MMS messages or delivery notifications in the Common Message Store by matching it with the Message-ID contained in the Message-Correlator of each stored messages.

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 4.1.11.

### 4.1.10 Correlation Algorithm for SMS

In order to ensure the message correlation algorithm succeeds on both the client and Message Store Server, RCS clients or the entity in the network sending SMS messages with characters from the GSM 7 bit national single shift and locking shift tables, shall instead use UCS2 (2-byte Universal Character Set) encoding.

NOTE: The RCS client may still receive SMS messages with characters from the GSM 7 bit national single shift and locking shift tables since they may come from non-RCS compliant clients or networks.

When other alphabets (e.g. Latin-1, HP Roman-8) are used as consequence of the SMS bearer technology in the network such as SMPP, the entity creating the Message-Correlator converts the message from the received alphabet into UTF-8 format. The entity which converts from UCS2 or GSM 7 bit alphabets to other alphabets (e.g. Latin-1, HP Roman-8), or vice versa, shall ensure a one-to-one character mapping. This entity can be the entity that stores the message and creates the Message-Correlator, or it can be the entity that sends the message towards the recipient.

NOTE: The one-to-one character mapping is vendor dependent when anything other than GSM 7 bit default alphabet or UCS2 is received by the SMS-C.

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 Message-Correlator 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 Message-Correlator header value as follows:

  - For messages with no text payload in the SMS user data a Message-Correlator header with no value shall be generated.
  - The entity creating the Message-Correlator converts from its original encoding (GSM 7 bit default alphabet or UCS2, see [3GPP TS 23.038], or any other GSM 7 bit national single shift and locking shift tables) into UTF-8 format. The same applies when alphabets other than GSM 7 bit or UCS2 are used (e.g. Latin-1, HP Roman-8).
  - Any UTF-8 "Null" character is removed.
  - Any UTF-8 characters "CR" and "LF", and the sequence "CR LF", are all removed
  - In the case of concatenated SMS messages once the message is reassembled and the above rules have been applied, only the first 160 characters shall be used to generate the Message-Correlator header value in accordance with the procedures defined above.
  - If the resulting string contains only US-ASCII characters (0x20 – 0x7e) it will be taken as the value of the Message-Correlator header.

- If the resulting string contains at least one non US-ASCII character, the Message-Correlator 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 Message-Correlator header encoding refer to [CPM-MSGSTOR-REST].
- Examples of Message-Correlator header values:
  the Message-Correlator header value of a short message with the text payload:
  `To your health, my friend`
  will encoded as follows
  `Message-Correlator: To your health, my friend`
  the Message-Correlator header of a short message with the text payload
  `На здоровье, мой друг`
  will be encoded as follows
  `Message-Correlator: =?utf-8?b?`
  `0J3QsCDQt9C00L7RgNC+0LLRjNC1LCDQvNC+0Lkg0LTRgNGD0LM=?=`

Table 74 illustrates the required coding conversions for the Message-Correlator algorithm to succeed.

| ID | Messaging Technology | Original Encoding type | Client Sender | | Client Receiver | Messaging Server is Receiver from SMS-C via SMPP or MAP |
|---|---|---|---|---|---|---|
| | | | Encoding for Sending Message | Calculation of Message-Correlator | Calculation of Message-Correlator | Calculation of Message-Correlator |
| 1 | Legacy SMS | GSM 7 bit default alphabet | GSM 7 bit default alphabet | Convert original message text payload into UTF-8 | Convert received message text payload into UTF-8 | Convert received message text payload into UTF-8 |
| 2 | | UCS2 | UCS2 | Convert original message text payload into UTF-8 | Convert received message text payload into UTF-8 | Convert received message text payload into UTF-8 |

| ID | Messaging Technology | Original Encoding type | Client Sender | | Client Receiver | Messaging Server is Receiver from SMS-C via SMPP or MAP |
|---|---|---|---|---|---|---|
| | | | Encoding for Sending Message | Calculation of Message-Correlator | Calculation of Message-Correlator | Calculation of Message-Correlator |
| 3 | | Other alphabets using an 8 bit encoding (e.g. Latin-1, HP Roman-8) | Not applicable [1] | Not applicable[1] | Not applicable[1] | Convert received message text payload into UTF-8.<br>The entity which converts from UCS2 or GSM 7 bit alphabets to other alphabets (e.g. Latin-1, HP Roman-8) shall ensure a one-to-one character mapping |
| 4 | | Message with GSM 7 bit national alphabet (single shift and locking shift tables) | UCS2 | Convert original message text payload into UTF-8 | Convert received message text payload into UTF-8 | Convert received message text payload into UTF-8 |
| 5 | | Message with GSM 7 bit national alphabet (single shift and locking shift tables) | GSM 7 bit (including shift tables[2]) (used by legacy SMS clients that are not RCS compliant) | Original message to UTF-8 using the GSM 7 bit national single shift and locking shift tables[2] | Convert received message text payload into UTF-8 using the GSM 7 bit national single shift and locking shift tables[2] | Convert received message text payload into UTF-8.<br>Support of GSM 7 bit national single shift and locking shift table is required by the entity generating the Message-Correlator.<br>NOTE: if a network entity is involved which converts this to another alphabet (e.g. Latin-1, HP Roman-8), a one-to-one mapping may not be possible and thus the Message-Correlator algorithm would not succeed. |

| ID | Messaging Technology | Original Encoding type | Client Sender | | Client Receiver | Messaging Server is Receiver from SMS-C via SMPP or MAP |
|---|---|---|---|---|---|---|
| | | | Encoding for Sending Message | Calculation of Message-Correlator | Calculation of Message-Correlator | Calculation of Message-Correlator |
| 6 | Standalone Message (when stored as SMS) | UTF-8 | UTF-8 | When stored as SMS, use original message text payload which is already in UTF-8 | When stored as SMS, use original message text payload which is already in UTF-8 | Convert received message text payload into UTF-8.<br>NOTE: For standalone messages with Pager Mode, it is possible based on Service Provider policies, for these messages to be automatically stored in the CMS as legacy SMS messages. In this case the client will not be able to correlate these messages with their copy in the Common Message Store based on the procedures for matching of messages defined in [RCS-CPM-CONVFUNC-ENDORS]. Since the client is not able to correlate the standalone messages with Pager Mode based on the procedures defined in [RCS-CPM-CONVFUNC-ENDORS], it shall then attempt to correlate them as per this row, i.e. using Message-Correlator. See section 4.1.16.7. |

**Table 74: Encoding conversions for Message-Correlator algorithm**

NOTES to Table 74:

1. A client encoding SMS messages using 8 bit encoding alphabets (e.g. Latin-1, HP Roman-8) instead of GSM 7 bit or UCS2 or a client sending encrypted messages over SMS are not supported by the current algorithm.
2. Correlation of SMS messages using national single shift and locking shift tables indicated in the user data header Info Element is not fully supported, especially if either the client or the entity in the network has no access to the SMS PDU or does not support the used national single shift or locking shift tables. Therefore an RCS Client shall encode SMS messages as UCS2.

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 Message-Correlator 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 Message-Correlator 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 or if the message contained characters encoded via single or locking shift tables may compensate for this by using alternative matching algorithms which are out of scope for this specification.

### 4.1.11 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 synchronises 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 synchronisation. 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 Message-Correlator header but with UIDs x and y, $x < y$, and the device has received two messages with the same value C for the Message-Correlator 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 synchronisation 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 Message-Correlator 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 Message-Correlator header which it will then identify with UID y. Similarly, if the Common Message Store only has UID x producing value C for the Message-Correlator header but the device has both t1 and t2, the device should identify t1 with message x and expect a subsequent synchronisation 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.

### 4.1.12 Recording of SMS messages

Apart from text transfer SMS provides a number of enhanced messaging capabilities as well as device and service control functions. For the synchronisation of SMS messages across devices it is essential that the client and the Common Message Store assume a common rule for SMS message recording.

Short messages are recorded in the Common Message Store either by the network or by the device based on the definitions in section 4.1.8.1. Entities recording short messages need to follow these guidelines.

NOTE: If the client has no access to the SMS PDUs, it may assume that all locally received messages that it has access to fulfil the rules for recording.

Clients matching received messages with a message in the Common Message Store need to consider these guidelines to apply matching only for messages that are subject to recording.

For clients fetching messages from the Common Message Store to update the local storage no special considerations are required.

The following PDU types need to be recorded for mobile originated SMS:

- SMS-SUBMIT
- SMS-STATUS-REPORT

The following PDU types need to be recorded for mobile terminated SMS:

- SMS-DELIVER

For a definition of the SMS PDU types refer to [3GPP TS 23.040].

#### 4.1.12.1 Recording of SMS-SUBMIT and SMS-DELIVER

A short message may consist of multiple parts, the content transferred in the actual user data and the content of the user data headers inserted in the user data. The following sections define the recording rules for the two parts of the message.

##### 4.1.12.1.1 User Data

The User Data is the field of a Short Message PDU that carries the user content (see [3GPP TS 23.040]). This section provides the rules for recording of messages based on the values of the SMS Data Coding Scheme and Protocol Identifier fields [3GPP TS 23.040].

##### 4.1.12.1.2 SMS Data Coding Scheme

The SMS Data Coding Scheme field of the short message indicates the encoding used for the user data.

The indication of text compression and character set are used by the recording entity to decode the message content. The message is not stored in the Common Message Store if the Character Set indicates "8 bit data". In all other cases the message is converted in text using the UTF-8 character set for storage. If the User Data is the only content of an SMS message it is stored in the Common Message Store the body of a CPIM message.

The SMS Class determines the routing of a mobile terminated short message on the device. Messages without a Message Class shall be recorded based on the principles defined in section 4.1.8.

A mobile terminated message indicating Class "0" shall not be recorded in the network. It is sent to the mobile device via SMS. The device will display the message immediately without

storing. If the user decides to store the message locally on the device after display the client shall upload the message to the Common Message Store without applying the Correlation Mechanism defined in section 4.1.8. The client shall store the message under the Default folder identified by the identity of the Contact in the Conversation. If this folder is not yet created in the Default folder, the client shall first create it. The client shall store the message with a Message-Context header set to pager-message as described in section 4.1.8.

Mobile terminated messages with Message Class "1", "2" and "3" shall be recorded by the network or the client unless content encoding (e.g. "8 bit data") or the high layer protocol indication prohibit recording.

If the Data Coding Scheme indicates a message for automatic deletion after reading it shall not be recorded.

Messages with a Message Waiting Indication shall not be recorded.

#### 4.1.12.1.3 Protocol Identifier

The Protocol Identifier indicates whether a higher layer protocol is used to encode the content of the User Data. Only messages with a Protocol Identifier set to the default 0x00 and to values from the range 0x81-0x87 (Replace Short Message Type) shall be recorded.

If a Replace Short Message Type is set in the Protocol Identifier of a mobile terminated short message (value in the range of 0x81-0x87), then the client shall apply the handling as defined in [3GPP TS 23.040], i.e. it replaces the content of a stored message that matches the Replace Type of the new message. If the client configuration parameter SMS MESSAGE STORE is set to value "1" or "2" the client shall update the Common Message Store according to the result of the local message processing. If the client stores a message in the Common Message the Replace-Short-Message-Type header value shall be set to the value received in the short message.

The entity in the network storing the message in the Common Message Store shall apply the following handling. If a message with a Replace Short Message Type is received, it shall select the folder where the conversation is stored, as identified by the originator address. It shall search in the folder the messages with the same Replace Short Message Type as the new message.

If a message with the same Short Message Replace Type header is found, the \Deleted flag shall be set for this message. The new message is stored in the selected folder with a Short Message Replace Type header set.

If no related message or folder exists the message is stored in the regular fashion. The Message is always stored with the Replace-Short-Message-Type header value set to the value received in the short message.

#### 4.1.12.1.4 Short Message Object

A short message shall be recorded in the Common Message Store with the headers and attributes set as defined below.

| Attribute | Status | Content |
|---|---|---|
| From | Mandatory | For mobile originated Short Messages it contains the public user identity of the sender. It is either encoded in the global-number representation of a tel URI as defined in [RFC3966] or as a SIP URI as defined in [RFC3261].<br>For mobile terminated Short Messages the value is derived from the SMS originator address.<br>If the originator is identified by a E.164 number it is encoded in the global-number representation of a tel URI as defined in [RFC3966]<br>If the originator is identified by a non E.164 number then it is encoded in the local-number representation of a tel URI as defined in [RFC3966] without parameters and context.<br>If the originator is identified by an alphanumeric string, the From field contains the string only. |
| To | Mandatory | For mobile originated short messages the value is derived from the SMS destination address.<br>If the destination is identified by a E.164 number it is encoded in the global-number representation of a tel URI as defined in [RFC3966]<br>If the destination is identified by a non E.164 number then it is encoded in the local-number representation of a tel URI as defined in [RFC3966] without parameters and context.<br>For mobile terminated Short Messages it contains the public user identity of the sender. It is encoded in the global-number representation of a tel URI as defined in [RFC3966] or as a SIP URI as defined in [RFC3261]. |
| Date | Mandatory | Indicates the time the message was recorded. Date format is according to [CPM-MSGSTOR-REST] |
| Conversation-ID | Mandatory | It shall be assigned by the entity that stores the message |
| Contribution-ID | Mandatory | It shall be assigned by the entity that stores the message. |
| Correlation-ID | Mandatory | IMDN Message ID will be the Object ID of the IMDN within the Message Store and assigned by the store. The IMDN is correlated to the original message using the correlation ID of that CPM message. |
| Correlation-Tag | Optional | If present it shall contain the SMS message Correlator for message correlation as defined in section 4.1.9. See also section 4.1.8. |
| Message-Context | Mandatory | Message-Context shall be set to "pager-message" |
| Direction | Optional | Direction attribute value shall be set as defined in section 4.1.16.8. |
| Message-ID | Optional | It is assigned by the entity that stores the message. If stored, the header value shall conform to the definitions of [RFC5322]. |
| Replace-Short-Message-Type | Optional | Indicates the replacement type of the short message. It can have the values 1 – 7. The value shall be taken from the value of SMS Protocol Identifier (see [3GPP TS 23.040]) |
| Content-Type | Mandatory | Message/CPIM |

| Attribute | Status | Content |
|---|---|---|
| CPIM | Mandatory | The attribute “CPIM” contains the entire CPIM message headers as a single string. The other fields and attributes are as defined in Table 76 below. In particular, the Content-Type attribute is the content type of the encapsulated MIME message body. The payload(s) of the message object are the body part(s) of the encapsulated MIME message body of the CPIM message. |

**Table 75: Object Attributes of the Short Message Object**

The CPIM attribute of the Short Message Object shall contain the CPIM headers and their values defined in Table 76. The message body shall the contain the content only.

| CPIM Header | Status | Content |
|---|---|---|
| From | Mandatory | For mobile originated Short Messages it contains the public user identity of the sender. It is either encoded in the global-number representation of a tel URI as defined in [RFC3966] or as a SIP URI as defined in [RFC3261].<br><br>For mobile terminated Short Messages the value is derived from the SMS originator address.<br>If the originator is identified by a E.164 number it is encoded in the global-number representation of a tel URI as defined in [RFC3966]<br>If the originator is identified by a non E.164 number then it is encoded in the local-number representation of a tel URI as defined in [RFC3966] without parameters and context.<br>If the originator is identified by an alphanumeric string, the From field contains the string only. |
| To | Mandatory | For mobile originated short messages the value is derived from the SMS destination address.<br>If the destination is identified by a E.164 number it is encoded in the global-number representation of a tel URI as defined in [RFC3966]<br>If the destination is identified by a non E.164 number then it is encoded in the local-number representation of a tel URI as defined in [RFC3966] without parameters and context.<br><br>For mobile terminated short messages it contains the public user identity of the sender. It is encoded in the global-number representation of a tel URI as defined in [RFC3966] or a SIP URI as defined in [RFC3261]. |

| CPIM Header | Status | Content |
|---|---|---|
| DateTime | Mandatory | For Mobile Terminated Messages it should indicate the time the message was received in the Originator Service Centre. Derived from the SMS Service Centre Time Stamp (see [3GPP TS 23.040]). If the SMS Service Centre Timestamp is not available it should contain the time of message recording.<br><br>For Mobile Originated Messages it should indicate the time the message was sent by the client that recorded the message or the time the message was received in the Originator Service Centre. |
| rcs.Message-Correlator | Optional | If present it shall contain the SMS message Correlator for message correlation as defined in section 4.1.9. See also section 4.1.8. |
| rcs.Message-Context | Optional | The value of rcs.Message-Context shall be set to "pager-message". It shall be present if the rcs.Message-Correlator header is present. It indicates that the rcs.Message-Correlator contains the SMS message correlation as defined in section 4.1.9. |
| rcs.Service-Centre-Address | Optional | Indicates the address of the Short Message Service Centre used for the transfer of the short message. For mobile originated messages it is derived from the transport destination address. For mobile terminated messages it is derived from the transport originating address. See [3GPP TS 23.040] |
| rcs.Reply-Path | Optional | Indicates whether a reply-path exists for the message. If the value is set to "1" the reply-path exists, if set to "0" or the header is not present the reply-path does not exist. For a description of the reply-path refer to [3GPP TS 23.040]. |
| rcs.Replace-Short-Message-Type | Optional | Indicates the replacement type of the short message. It can have the values 1 – 7. The value shall be taken from the value of SMS Protocol Identifier (see [3GPP TS 23.040]) |
| Content-Type | Mandatory | For messages with text part only the Message body should be encoded using content-type text/plain in UTF-8 encoding. |
| Content-Transfer-Encoding | Optional | Typical content transfer encoding shall be used, e.g. quoted-printable or base64 |

**Table 76: CPIM Headers of the Short Message Object**

Example of a recorded short message:

```
POST http://nms-
sib01.si.enclab.MNO.net/nms/v1/base/tel:+19717774171/objects HTTP/1.1
Accept-Encoding: gzip,deflate
MIME-Version: 1.0
Accept: application/json
Authorization: Bearer
PAT_bceNdO3GSKwkHBkL1rnEceW2TTLx3ijGzbonv3qFvVuYHnocDopEZ2eyMfzea/2POmQrLn
YJch2VDeEA2ooRh4o5RoiNxpUVvp+e4P+GDYfkL8+5SwGYjVIBhO+AaDS8wWxYA4lA/qk9lUO+
DEhCcp+JHP2QDqvhvschz0ZUesIOe/cLgAQpMhD2k2mOhtL3BO6N5zPm1rBLeM8QjaMHKfdoDg
BnWBFKTf+NGlHwbutkX2ZAfApQ4VCL7wryWkuBRNpWVHacFfjC+kxJy+IYdno5VC2yxEj3l9Mp
gJgpEpbSAf5JtpMhBX+Il+It9vG8
```

```
Content-Type: multipart/form-data; boundary=ZQ87HPOZX.OBADLDBLD
Content-Length: 951
Host: nms-sib01.si.enclab.MNO.net
Proxy-Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

--ZQ87HPOZX.OBADLDBLD
Content-Disposition: form-data; name=root-fields
Content-Type: application/json; charset=UTF-8

{
  "object": {
    "attributes": {
      "attribute": [{
        "name": "Subject",
        "value": ["Test Object"]
      }, {
        "name": "From",
        "value": ["+19995551212"]
      }, {
        "name": "Date",
        "value": ["2015-04-01T14:30:30Z"]
      }, {
        "name": "Message-Context",
        "value": ["pager-message"]
      }, {
        "name": "To",
        "value": ["+19717774171"]
      }, {
        "name": "Direction",
        "value": ["In"]
      }, {
        "name": "CPIM",
        "value": ["From:+19995551212\r\nTo:+19717774171\r\nDate:2015-04-
               01T14:30:30Z\r\nrContent-Type:text/plain\r\nNS:rcs
               <http://www.gsma.com>\r\nrcs.Message-Context:pager-
               message"]
      }]
    },
    "flags": { "flag": []}
  }
}
--ZQ87HPOZX.OBADLDBLD
Content-Disposition: form-data; name=message
Content-Type: multipart/mixed; boundary=EFO94Y8QA.T8UCA0REW

--EFO94Y8QA.T8UCA0REW
Content-Disposition: attachment; filename=sms; name=sms
Content-Type: text/plain

this is a test SMS message
--EFO94Y8QA.T8UCA0REW--
--ZQ87HPOZX.OBADLDBLD—
```

#### 4.1.12.1.5 User Data Header

User Data Headers can be used for encoding for SMS control functions as well as special SMS content (see [3GPP TS 23.040]. The type of content of a User Data Header is identified by the Information Element Identifier (IEI) within a User Data header. The following Information Identifiers need to be considered for the processing of short messages.

Concatenated short messages reference numbers IEI need to be considered when processing a concatenated message. Only the re-assembled message shall be recorded in the Common Message Store.

NOTE: The recording requirements for other values of Information Element Identifiers are for further study.

### 4.1.12.2 Recording of SMS-STATUS-REPORT

The SMS-STATUS-REPORT informs the message sender about the status of a previously sent message if it has been requested by the sender. The SMS-STATUS-REPORT shall be used by the entity storing SMS messages in the Common Message Store to record a delivery notification as defined in [CPM-MSGSTOR-REST].

The SMS-STATUS-REPORT is matched on the originating side to the original sent message by use of SMS Message Reference assigned by the originating device. The SMS Message Reference is not globally unique, thus the matching of SMS-STATUS-REPORT to the original short message across multiple devices has some limitations.

A delivery notification shall be recorded if the Status field in the SMS-STATUS-REPORT indicates that the short message has been "received by the SME" (see [3GPP TS 23.040]). In all other cases a delivery notification shall not be recorded.

A SMS delivery notification shall be recorded in the Common Message Store with Object Attributes and CPIM headers set as follows.

| Attribute | Status | Content |
|---|---|---|
| From | Mandatory | It contains the address of the recipient of the original message as derived from the recipient address field of the SMS-STATUS-REPORT see [3GPP TS 23.040].<br>If identified by a E.164 number it is encoded in the global-number representation of a tel URI as defined in [RFC3966]<br>If identified by a non E.164 number then it is encoded in the local-number representation of a tel URI as defined in [RFC3966] without parameters and context. |
| To | Mandatory | It contains the public user identity of the sender. It is encoded in the global-number representation of a tel URI as defined in [RFC3966] or as a SIP URI as defined in [RFC3261]. |
| Date | Mandatory | Indicates the time the message was recorded |
| Conversation-ID | Mandatory | It shall have the same value as the original sent message. |
| Contribution-ID | Mandatory | It shall have the same value as the original sent message. |

| Attribute | Status | Content |
|---|---|---|
| Message-ID | Optional | It is assigned by the entity that stores the message. If stored, the header value shall conform to the definitions of [RFC5322]. |
| Content-Type | Mandatory | Message/CPIM |
| CPIM | Mandatory | The attribute “CPIM” contains the entire CPIM message headers as a single string. The other fields and attributes are as defined in Table 78 below. In particular, the Content-Type attribute is the content type of the encapsulated MIME message body. The payload(s) of the message object are the body part(s) of the encapsulated MIME message body of the CPIM message |

**Table 77: Object Attributes of the SMS Delivery Notification**

The CPIM attributes of the SMS Delivery Notification shall contain the CPIM headers and their values defined in Table 78:

| CPIM Header | Status | Content |
|---|---|---|
| From | Mandatory | It contains the address of the recipient of the original message as derived from the recipient address field of the SMS-STATUS-REPORT see [3GPP TS 23.040].<br>If identified by a E.164 number it is encoded in the global-number representation of a tel URI as defined in [RFC3966]<br>If identified by a non E.164 number then it is encoded in the local-number representation of a tel URI as defined in [RFC3966] without parameters and context. |
| To | Mandatory | It contains the public user identity of the sender. It is encoded in the global-number representation of a tel URI as defined in [RFC3966] |
| DateTime | Mandatory | It should indicate the time the message was delivered by the Service Centre. The value is derived from the Discharge Time of the SMS-STATUS-REPORT (see [3GPP TS 23.040]). If it is not available then it should contain the time of message recording. |
| Content-Type | Optional | message/imdn+xml |

**Table 78: CPIM Headers of the SMS Delivery Notification**

The message body of the CPIM message shall contain an IMDN with status set to "delivered" and the imdn.Message-ID assigned to the original sent message.

### 4.1.13 Recording of MMS messages

MMS provides a rich messaging service for multimedia content to be sent to a single recipient or to a list of recipients.

Multimedia messages are recorded in the Common Message Store either by the network or by the device based on the definitions in section 4.1.8.1. Entities recording multimedia messages need to follow these guidelines.

The following PDU types need to be recorded for mobile originated MMS:

- MM1 Submission
- MM1 Delivery Report

- MM1 Read-Reply Report

The following PDU types need to be recorded for mobile terminated MMS:

- MM1 Delivery
- MM1 Read-Reply Report

#### 4.1.13.1 Recording of MM1 Submission and MM1 Delivery

The entity storing a MMS message in the Common Message Store shall use the Multimedia Message object defined in this section.

A multimedia message shall be recorded in the Common Message Store with the headers set as defined below.

Note: The encoding of addresses in Object Attributes headers of a recorded MMS message follows the SIP encoding principles as defined in [CPM-MSGSTOR-REST] whereas the address encoding of an MMS is based on the MMS Addressing Model of [MMSENC]. The entity storing a MMS message needs to re-format address headers.

For an MMS Message to multiple recipients the entity storing the message shall store the participant list in a recipient-list-history content part with *copyControl* set in accordance with the recipient address type, To:, Cc: or Bcc:.

| Attribute | Status | Content |
|---|---|---|
| From | Mandatory | For mobile originated MMS messages it contains the public user identify of the sender. It is either encoded in the global-number representation of a tel URI as defined in [RFC3966] or as a SIP URI as defined in [RFC3261].<br>For mobile terminated MMS Messages the value is derived from the MMS originator address.<br>If the originator is identified by a E.164 number it is encoded in the global-number representation of an tel URI as defined in [RFC3966]<br>If the originator is identified by a non E.164 number then it is encoded in the local-number representation of a tel URI as defined in [RFC3966] without parameters and context.<br>If the originator is identified by an e-mail address it is encoded as mailbox as defined in [RFC5322]. |

| Attribute | Status | Content |
|---|---|---|
| To | Optional | For mobile originated MMS messages the value is derived from the MMS recipient address "To" field.<br>If the recipient is identified by an E.164 number it is encoded in the global-number representation of an tel URI as defined in [RFC3966]<br>If the recipient is identified by a non E.164 number then it is encoded in the local-number representation of a tel URI as defined in [RFC3966] without parameters and context.<br>If the originator is identified by an e-mail address it is encoded as mailbox as defined in [RFC5322].<br>For a MMS message to multiple recipients the To header value shall be encoded as [RFC5322] address-list with each address value following the encoding rules above. In addition the multiple recipient addresses shall be represented in a body part of the message containing a recipient-list-history as defined in [RFC5365].<br>For mobile terminated MMS Messages it contains the public user identity of the recipient. It is encoded in the global-number representation of a tel URI as defined in [RFC3966] or as SIP URI as defined in [RFC3261].<br>At least one of To, Cc or Bcc header shall be present in the message. |
| Cc | Optional | For mobile originated MMS messages the value is derived from the MMS recipient address "Cc" field.<br>If the recipient is identified by an E.164 number it is encoded in the global-number representation of an tel URI as defined in [RFC3966]<br>If the recipient is identified by a non E.164 number then it is encoded in the local-number representation of a tel URI as defined in [RFC3966] without parameters and context.<br>If the originator is identified by an e-mail address it is encoded as mailbox as defined in [RFC5322].<br><br>For a MMS message to multiple recipients the Cc header value shall be encoded as [RFC5322] address-list with each address value following the encoding rules above. Multiple recipient addresses shall be represented in a body part of the message containing a recipient-list-history as defined in [RFC5365].<br><br>For mobile terminated MMS Messages it contains the public user identity of the sender. It is encoded in the global-number representation of a tel URI as defined in [RFC3966] or a SIP URI as defined in [RFC3261].<br>At least one of To, Cc or Bcc header shall be present in the message. |

| Attribute | Status | Content |
|---|---|---|
| Bcc | Optional | For mobile originated MMS messages the value is derived from the MMS recipient address "Bcc" field.<br>If the recipient is identified by an E.164 number it is encoded in the global-number representation of an tel URI as defined in [RFC3966]<br>If the recipient is identified by a non E.164 number then it is encoded in the local-number representation of a tel URI as defined in [RFC3966] without parameters and context.<br>If the originator is identified by an e-mail address it is encoded as mailbox as defined in [RFC5322].<br><br>For a MMS message to multiple recipients the Bcc header value shall be encoded as [RFC5322] address-list with each address value following the encoding rules above. Multiple recipient addresses shall be represented in a body part of the message containing a recipient-list-history as defined in [RFC5365].<br><br>For mobile terminated MMS Messages it contains the public user identity of the sender. It is encoded in the global-number representation of a tel URI as defined in [RFC3966] or a SIP URI as defined in [RFC3261].<br>At least one of To, Cc or Bcc header shall be present in the message. |
| Date | Mandatory | Indicates the time the message was recorded |
| Conversation-ID | Mandatory | It shall be assigned by the entity that stores the message |
| Contribution-ID | Mandatory | It shall be assigned by the entity that stores the message. |
| Message-ID | Optional | It is assigned by the entity that stores the message. . If stored, the header value should conform to the definitions of [RFC5322].<br>In a transition period Service Providers may store in the Message-ID header the value of the MMS Message-ID for compatibility with earlier versions of this specification. |
| Message-Correlator | Mandatory | It shall contain the MMS Message-ID for message correlation as defined in section 4.1.9. See also section 4.1.8 |
| Message-Context | Mandatory | Message-Context shall be set to "multimedia-message" |
| Direction | Optional | Direction attribute value shall be set as defined in section 4.1.16.8. |
| Content-Type | Mandatory | Message/CPIM |
| CPIM | Mandatory | The attribute “CPIM” contains the entire CPIM message headers as a single string. The other fields and attributes are as defined in Table 80 below. In particular, the Content-Type attribute is the content type of the encapsulated MIME message body. The payload(s) of the message object are the body part(s) of the encapsulated MIME message body of the CPIM message. |

**Table 79: Object Attributes of the Multimedia Message Object**

The CPIM attribute of the Multimedia Message Object shall contain the CPIM headers and their values defined in Table 80. The body of the Multimedia Message Object shall contain the content only.

| CPIM Header | Status | Content |
|---|---|---|
| From | Mandatory | For mobile originated MMS messages it contains the public user identity of the sender. It is either encoded in the global-number representation of a tel URI as defined in [RFC3966] or as a SIP URI as defined in [RFC3261].<br><br>For mobile terminated MMS Messages the value is derived from the MMS originator address.<br>If the originator is identified by a E.164 number it is encoded in the global-number representation of an tel URI as defined in [RFC3966]<br>If the originator is identified by a non E.164 number then it is encoded in the local-number representation of a tel URI as defined in [RFC3966] without parameters and context.<br>If the originator is identified by an e-mail address it is encoded as mailbox as defined in [RFC5322]. |
| To | Optional | For mobile originated MMS messages the value is derived from the MMS recipient address "To" field.<br>If the recipient is identified by an E.164 number it is encoded in the global-number representation of an tel URI as defined in [RFC3966]<br>If the recipient is identified by a non E.164 number then it is encoded in the local-number representation of a tel URI as defined in [RFC3966] without parameters and context.<br>If the originator is identified by an e-mail address it is encoded as mailbox as defined in [RFC5322].<br><br>For a MMS message to multiple recipients there shall be To and Cc header fields per recipient address. Bcc recipients addresses shall be contained in a To header field. If a client needs to represent the Bcc destination address classification it shall use the header from the mail headers instead. Multiple recipient addresses shall be represented in a body part of the message containing a recipient-list-history as defined in [RFC5365] with a mapping of the address header field type to copyControl parameter.<br>For mobile terminated Short Messages it contains the public user identity of the sender. It is encoded in the global-number representation of a tel URI as defined in [RFC3966] or a SIP URI as defined in [RFC3261].<br>At least one of To or Cc header shall be present in the message. |

| CPIM Header | Status | Content |
|---|---|---|
| Cc | Optional | For mobile originated MMS messages the value is derived from the MMS recipient address "Cc" field.<br>If the recipient is identified by an E.164 number it is encoded in the global-number representation of a tel URI as defined in [RFC3966].<br>If the recipient is identified by a non E.164 number then it is encoded in the local-number representation of a tel URI as defined in [RFC3966] without parameters and context.<br>If the originator is identified by an e-mail address it is encoded as mailbox as defined in [RFC5322].<br><br>For a MMS message to multiple Cc recipients there shall be a Cc header field per recipient address.<br>At least one of To or Cc header shall be present in the message. |
| DateTime | Mandatory | For Mobile Terminated Messages it should indicate the time the message was received in the Originator Service Centre. Derived from the MMS Date header (see [MMSENC]). If the Date header is not available it should contain the time of message recording.<br><br>For Mobile Originated Messages it should indicate the time the message was sent by the client that recorded the message or the time the message was received in the Originator Service Centre. |
| Subject | Optional | Indicates the Subject of the MMS message. It shall be taken from the Subject header of the MMS message. |
| imdn.Disposition-Notification | Optional | This header indicates whether the sender has requested a MMS read-reply report.<br>The entity recording the message shall derive the value from the MMS X-Mms-Read-Reply header defined in [MMSENC]. If the X-Mms-Read-Reply header is present and set to "yes", the imdn.Disposition-Notification header shall take the value "display". Otherwise the header shall be absent. |
| rcs.Message-Correlator | Mandatory | It shall contain the MMS Message-ID. |
| rcs.Message-Context | Mandatory | The value of rcs.Message-Context shall be set to "multimedia-message". It indicates that the rcs.Message-Correlator contains the MMS message correlation (MMS Message-ID) as defined in section 4.1.9. |
| rcs.Mms-Message-Class | Optional | The MMS Message Class indicates the class of the multimedia message. It may take the values "Personal", “Advertisement", "Informational" and "Auto". Multimedia messages with value "Auto" typically contain MMS read report.<br>If rcs.Mms-Class is absent the value "Personal" shall be assumed. |

| CPIM Header | Status | Content |
|---|---|---|
| Content-Type | Mandatory | The Content-Type defines the type of content of the body part of the CPIM message which is likely to be a multipart.<br>For the user content the Content-Type shall be taken from the original multimedia message. All types of content allowed for MMS are relevant for storage in the Common Message Store, e.g. presentation, text, audio and video.<br>For definitions of the MMS Message Body structure refer to [MMSENC].<br>For a resource list body the content type as defined in [RFC4826] shall be used. |

**Table 80: CPIM Headers of the Multimedia Message Object**

Example of a recorded two part multimedia message:

```
POST http://nms-
sib01.si.enclab.MNO.net/nms/v1/base/tel:+19717774171/objects HTTP/1.1
Accept-Encoding: gzip,deflate
MIME-Version: 1.0
Accept: application/json
Authorization: Bearer
PAT_bceNdO3GSKwkHBkL1rnEceW2TTLx3ijGzbonv3qFvVuYHnocDopEZ2eyMfzea/2POmQrLn
YJch2VDeEA2ooRh4o5RoiNxpUVvp+e4P+GDYfkL8+5SwGYjVIBhO+AaDS8wWxYA4lA/qk9lUO+
DEhCcp+JHP2QDqvhvschz0ZUesIOe/cLgAQpMhD2k2mOhtL3BO6N5zPm1rBLeM8QjaMHKfdoDg
BnWBFKTf+NGlHwbutkX2ZAfApQ4VCL7wryWkuBRNpWVHacFfjC+kxJy+IYdno5VC2yxEj3l9Mp
gJgpEpbSAf5JtpMhBX+Il+It9vG8
Content-Type: multipart/form-data; boundary=I937053OH.VXS8GY5VX
Content-Length: 1022
Host: nms-sib01.si.enclab.MNO.net
Proxy-Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

--I937053OH.VXS8GY5VX
Content-Disposition: form-data; name=root-fields
Content-Type: application/json; charset=UTF-8

{
  "object": {
    "attributes": {
      "attribute": [{
        "name": "Subject",
        "value": ["Test Object"]
      }, {
        "name": "From",
        "value": ["+19995551212"]
      }, {
        "name": "Date",
        "value": ["2015-04-01T14:30:30Z"]
      }, {
        "name": "Message-Context",
        "value": ["multimedia-message"]
      }, {
```

```
        "name": "To",
        "value": ["+19717774171"]
      }, {
        "name": "Direction",
        "value": ["In"]
            }, {
        "name": "CPIM",
        "value": ["From:+19995551212\r\nTo:+19717774171\r\nDate:2015-04-
        01T14:30:30Z\r\nrContent-Type:multipart/mixed\r\nrcs.Message-
        Correlator:6HUU0UN9C.C4F0GD8SR\r\nrcs.Message-Context :multimedia-
        message\r\nNS:rcs <http://www.gsma.com>"]
        }]
    },
    "flags": { "flag": []},
    "correlationId": "6HUU0UN9C.C4F0GD8SR"
  }
}
--I937053OH.VXS8GY5VX
Content-Disposition: form-data; name=message
Content-Type: multipart/mixed; boundary=PFLOG1DC7.NVJS9F8UL

--PFLOG1DC7.NVJS9F8UL
Content-Disposition: attachment; filename=body; name=body
Content-Type: text/plain

this is the text body of a multimedia message
--PFLOG1DC7.NVJS9F8UL--
--I937053OH.VXS8GY5VX--
```

#### 4.1.13.2 Recording of MM1 Delivery Report

The entity storing a mobile originated MMS messages in the Common Message Store shall also store MM1 Delivery Reports.

A MM1 Delivery Report is matched on the originating side to the original sent message by use of MMS message-id value. In case of multiple recipients the address of the recipient from which the delivery report has been received needs to be taken into account.

The MM1 Delivery Indication shall be used by the entity storing MMS messages in the Common Message Store to record a Delivery Notification as defined in [CPM-MSGSTOR-REST].

A delivery notification shall be recorded if the MMS Status field in the MMS delivery report indicates that the short message has been "retrieved" (see [MMSENC]). In all other cases a delivery notification shall not be recorded.

A MMS delivery notification shall be recorded in the Common Message Store with Object Attributes and CPIM headers set as follows.

| Attribute | Status | Content |
|---|---|---|
| From | Mandatory | It contains the address of the recipient of the original message as derived from the MMS deliver indication, see [MMSENC].<br>If identified by a E.164 number it is encoded in the global-number representation of a tel URI as defined in [RFC3966]<br>If identified by a non E.164 number then it is encoded in the local-number representation of a tel URI as defined in [RFC3966] without parameters and context.<br>If identified by an e-mail address it is encoded as mailbox as defined in [RFC5322]. |
| To | Mandatory | It contains the public user identity of the sender of the original message. It is encoded in the global-number representation of a tel URI as defined in [RFC3966] or as a SIP URI as defined in [RFC3261]. |
| Date | Mandatory | Indicates the time the message was recorded |
| Conversation-ID | Mandatory | It shall have the same value as the original sent message. |
| Contribution-ID | Mandatory | It shall have the same value as the original sent message. |
| Message-ID | Optional | It is assigned by the entity that stores the message. If stored, the header value should conform to the definitions of [RFC5322].<br>In a transition period Service Providers may store in the Message-ID header the value of the MMS Message-ID for compatibility with earlier versions of this specification. |
| Message-Correlator | Mandatory | It shall contain the MMS Message-ID for message correlation as defined in section 4.1.9. See also section 4.1.8. It has the same value as the Message-Correlator header of the original sent message. |
| Content-Type | Mandatory | Message/CPIM |
| CPIM | Mandatory | The attribute “CPIM” contains the entire CPIM message headers as a single string. The other fields and attributes are as defined in Table 82 below. In particular, the Content-Type attribute is the content type of the encapsulated MIME message body. The payload(s) of the message object are the body part(s) of the encapsulated MIME message body of the CPIM message. |

**Table 81: Object Attributes of the MMS Delivery Notification**

The CPIM attribute of the MMS Delivery Notification shall contain the CPIM headers and their values defined in Table 82:

| CPIM Header | Status | Content |
|---|---|---|
| From | Mandatory | It contains the address of the recipient of the original message as derived from the MMS deliver indication, see [MMSENC].<br>If identified by a E.164 number it is encoded in the global-number representation of a tel URI as defined in [RFC3966]<br>If identified by a non E.164 number then it is encoded in the local-number representation of a tel URI as defined in [RFC3966] without parameters and context.<br>If identified by an e-mail address it is encoded as mailbox as defined in [RFC5322]. |
| To | Mandatory | It contains the public user identity of the sender. It is encoded in the global-number representation of a tel URI as defined in [RFC3966] |
| DateTime | Mandatory | It should indicate the time the message was delivered by the Service Centre. The value is derived from the Date field of the MM1 Delivery report, see [MMSENC]. If it is not available then it should contain the time of message recording. |
| rcs.Message-Correlator | Mandatory | It shall contain the MMS Message-ID for message correlation as defined in section 4.1.9. See also section 4.1.8. It has the same value as the rcs.Message-Correlator header of the original sent message. |
| Content-Type | Optional | message/imdn+xml |

**Table 82: CPIM Headers of the MMS Delivery Notification**

The message body of the CPIM message shall contain an IMDN with status set to "delivered" and the imdn.Message-ID assigned to the original sent message.

#### 4.1.13.3 Recording of MMS Read Reports

According to [MMSCTR] there are two methods in MMS to transfer a Read Report from the recipient to the sender.

1. Multimedia Message Read Report:
   After display of a MMS message for which the originator has requested a read reply and the sending of a read report is authorised by the recipient user the client generates an "automatic" MMS message and sends it back to the sender. The entity storing MMS messages in the Common Message Store shall store the multimedia message read report message as a normal MMS message as defined in section 4.1.13.1. It is essential that the storing entity sets the value of the rcs.Mms-Message-Class header with the value received in the message, i.e. "auto".
   Clients fetching a MMS message and the related multimedia message read report from the Common Message Store should present it in the message history in the same way as if the message and the read report would have been received via MMS.
2. PDU Read Report:
   The implementation of recording of MMS PDU Read Reports is for further study.

A client sending a MMS Read Report to the sender shall not regard it as a display IMDN notification which is used by the messaging server participating functions as a trigger to set the \Seen flag in the Common Message Store. Therefore the client shall follow the

procedures for the setting of the \Seen flag in Common Message store defined in section 4.1.14 as if an IMDN display notification was not requested.

### 4.1.14 Optimisations for UNI operations to Common Message Store

If the MESSAGE STORE EVENT REPORTING configuration parameter is enabled (see section A.1.3), the RCS client shall use the CPM event reporting framework procedures (see section 6.7 of [RCS-CPM-CONVFUNC-ENDORS]) in an established 1-to-1 Chat session.

NOTE: Section 6.7 of [RCS-CPM-CONVFUNC-ENDORS] specifies the functionality as CPM IMAP events with a content type "*application/vnd.oma.cpm-event-imap+xml* ". The functionality can be used in combination with the RESTful based message store access used in this section though.

The RCS client shall use this for the following cases between RCS client and Participating Function, in order to report:

1. when one or more messages have been read by the RCS user, so that the Participating Function can set the “\Seen” flag for the stored message object(s) in CMS on behalf of the RCS client; and,
2. When a message was deleted by the RCS user, so that the Participating Function can set the “\Deleted” flag for the stored message object(s) in CMS.

To set the “\Seen” flag (case 1):

When the RCS user has read/displayed a received message, the RCS client shall inform the Common Message Store so that the message will be shown as “read” on other user devices after synchronizing with the Common Message Store.

This can be realised in the following ways:

1. If an IMDN display notification was requested for the received message and one was generated by the RCS client, the Participating Function (B2BUA) receiving the IMDN display notification from the client (via MSRP or SIP MESSAGE request) shall set the “\Seen” flag in CMS for the messages reported;
2. If an IMDN display notification was not requested, or if the RCS user settings on the client disabled sending them for read messages, then the CPM event reporting framework is used to report to the Participating Function that a message was read by the RCS client.

To set the “\Deleted” flag (case 2):

When the RCS user has deleted a received message, the RCS client shall inform the Common Message Store so that the message will be shown as “deleted” on other user devices after synchronizing with the Common Message Store. This is realised by the CPM event reporting framework being used to report to the Participating Function that a message was deleted by the RCS client.

### 4.1.15 A Common File Store for File Transfer via HTTP

#### 4.1.15.1 Overview

For File Transfer as defined in section 3.2.5, the storage for files/content transferred via File Transfer may be kept separate from the smaller, text based content and metadata. It also may be desirable to keep received content for longer than the original validity period. For these purposes, a Common File Store may be used together with a Common Message Store. The Common Message Store stores the chat messages carrying the File Transfer content body containing the meta-information and links for the thumbnail (optional) and the file(content), while the Common File Store stores the actual (i.e. local) copy of the thumbnail and/or content referred to by those URL links.

Similar to the Common Message Store, the Common File Store may be used to synchronise files between devices in the recipient's network in addition to keeping a back-up of files in the local network.

When the Common File Store is deployed, it shall behave towards the client as a HTTP Content Server supporting both sender procedures for File Transfer (see section 3.2.5.3.1) and receiver procedures for File Transfer (see section 3.2.5.3.2).

For the Common File Store to work with the Common Message Store a Service Provider deploying these shall support functions for File Transfer localisation in the terminating network as defined in section 4.1.15.3.

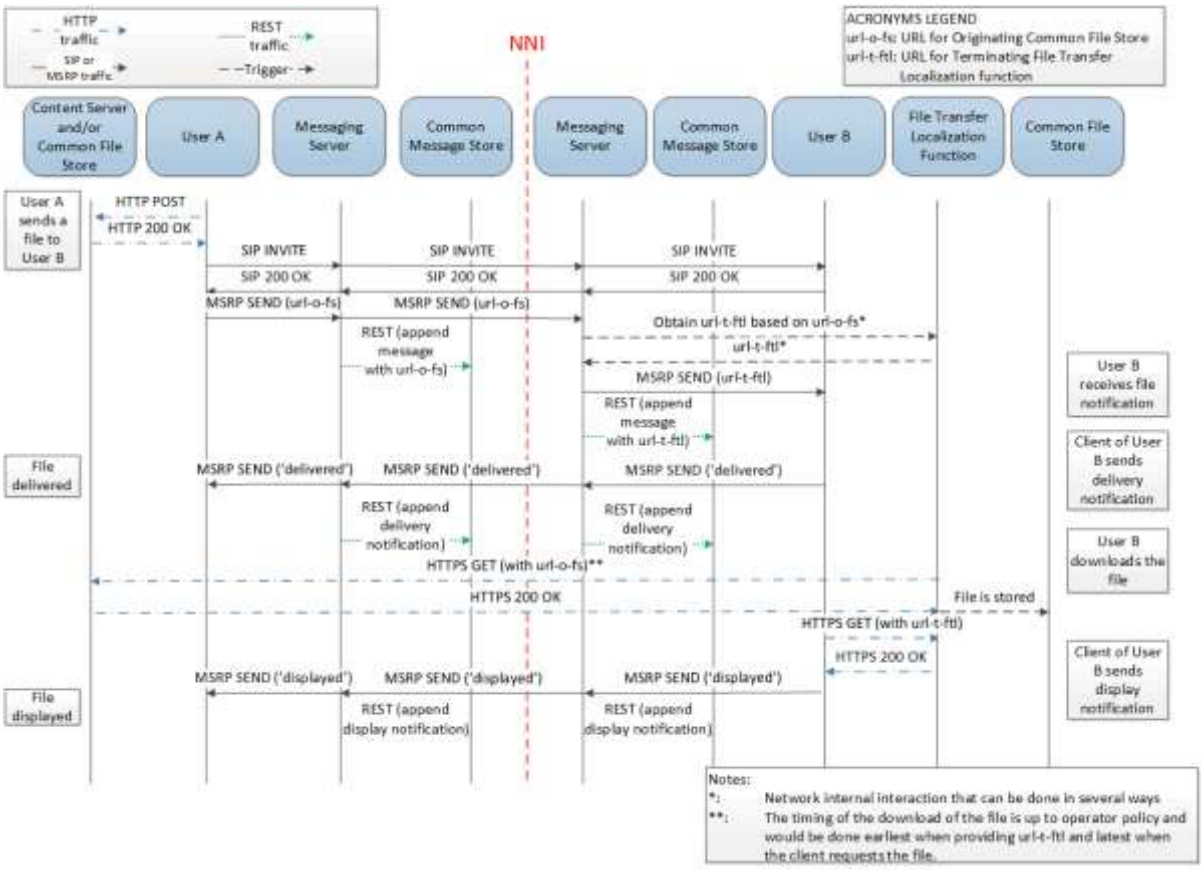



**Figure 18: File Transfer Localisation Function: global flow**

#### 4.1.15.2 Sender Procedures

The sender procedures defined in section 3.2.5.3.1 apply. The Service Provider shall direct the files uploaded by the sender towards a permanent storage with or without triggering the File Transfer Localisation Function. The Service Provider can assign a specific Content Server URI for different types of clients (e.g. legacy) via the configuration parameter FT HTTP CS URI defined in section A.1.4.

#### 4.1.15.3 Localisation in the terminating Service Provider's network

Localisation is the process to ensure that the terminating Service Provider is involved in the download of files referred to in the URLs conveyed using the messaging services. Localisation applies to the URLs

- for the file and thumbnail contained in a File Transfer body defined in section 3.2.5.4,
- for the media and thumbnail contained in a Rich Card's body as described in section 3.6.10.5.3,
- conforming to the definitions of the HTTP Content Server URL as defined in section 3.2.5.5 in the text body of a Standalone Message.

Once downloaded, the terminating Service Provider shall store files in the Common File Store.

Localisation requires

- a function to localise the download URL received via the messaging service by replacing the remote download URL by a local download URL. A Service Provider performing localisation shall apply localisation of the download URL via

  - the client-based local download of files as defined in section 3.2.5.3.2, and/or
  - procedures in the terminating messaging server as defined in section 4.1.15.3.1.

  The local download URL shall be resolvable to the remote download URL. It may be used to convey additional meta information.

  NOTE: If the terminating Service Provider relies on CFS as defined in section 3.2.3.8.1 and if the terminating Service Provider has no control over the content of SMS messages that are sent on File Transfer fall-back as defined in section 3.2.5.7.3.2, then the client-based local download of files approach can be used only.

- a function to process the local download of files in the network that

  - processes the client's request to download the file via the local download URL,
  - resolves the remote download URL,
  - downloads the file from the originating network and,
  - manages the local storage of the file in the Common File Store.

  This function is provided via the Localisation Function defined in section 4.1.15.3.3.

#### 4.1.15.3.1 Terminating Messaging Server Localisation Procedures

If the Service Provider applies localisation of the URL in the terminating Messaging Server, the terminating Messaging Server shall, when receiving a Chat, Group Chat, Chatbot or Standalone message, screen the message content for bodies containing download URLs.

If a relevant message body with URL(s) is found, then the Messaging Server shall either

- create the local download URL(s), or
- as a deployment option, invoke the Localisation Function to convey the remote download URL(s) to the Localisation Function, to obtain the local download URL(s) and optionally to update the validity of the file. The technical procedures for the trigger processing of the Messaging Server and the Localisation Function are out of the scope of this document.

The terminating Messaging Server shall update the message to contain the local download URL(s) and optionally the new validity of the file(s). The terminating Messaging Server shall record the message resulting from the processing in the Common Message Store.

#### 4.1.15.3.2 Receiver Procedures

The receiver procedures defined in section 3.2.5.3.2 apply. The procedures apply for cases where the client retrieves the file

- in result of reception of a Chat or Standalone Message with a File Transfer content body or any other content URL the client received via RCS messaging received from another user or sent by the own user as indicated via the CPIM "Message-Direction" header defined in section C.1.9 of [RCS-CPM-CONVFUNC-ENDORS],
- in result of retrieval of a message object with a File Transfer content body from the Common Message Store which is received from another user or sent by the own user as indicated via the "Direction" attribute defined in [CPM-MSGSTOR-REST].

Since the validity attribute of a file in the File Transfer message body indicates the validity for the first time download request, the client shall respect the value of the validity parameter only for messages received via Chat or Standalone Message which are not synchronised with the Common Message Store as defined in section 4.1.16.7. Once the client sends the file download request, the file is localised and made available for future retrievals. In the case where the Common File Store is deployed with the Common Message Store, for the XML in File Transfer messages retrieved from the Message Store Server, the validity in the XML is not relevant and should be ignored since the validity period configured for localised files in the Common File Store applies.

NOTE: When a Common File Store is deployed, it is up to service provider policy whether the received files are assumed to be archived if the Archived flag is set for the link to the file.

#### 4.1.15.3.3 Localisation Function Procedures

If the Localisation Function is invoked by the Messaging Server via the deployment option defined in section 4.1.15.3.1, then the Localisation Function shall interact accordingly, i.e. the Localisation Function

- shall provide the local download URL for a given remote download URL, and
- may provide a new validity for the file.

The Localisation Function may download the file from the originating network using the remote download URL the earliest when invoked by the Messaging Server, but latest when it receives a download request from the client, as per Service Provider policy. Once downloaded, the Localisation Function shall store the file in the Common File Store in the Service Provider network, in relation to the remote download URL.

If the Localisation Function receives a download request from the client, then the Localisation Function

- shall support the file download procedure defined in section 4.1.15.3.2,
- shall determine based on the provided URL whether the file is stored in the Common File Store in the Service Provider network
- if stored already, the Localisation Function shall provide the file from the Common File Store, otherwise
- the Localisation Function shall attempt to download the file from the originating network as defined in section 3.2.5.3.2. The Localisation Function shall relay responses received from the originating network to the client. If the file is downloaded successfully, then the Localisation Function shall store it in the Common File Store in the Service Provider network.

### 4.1.16 Client behaviour

Clients shall comply with the operations and procedures described in sections 5.5.2 of [CPM-SYS_DESC] and 6 of [CPM-MSGSTOR-REST]. Some further clarifications on the client expected behaviour when it interacts with the Message Store Server are presented in the following sections.

#### 4.1.16.1 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 6.4.1.3 of [CPM-MSGSTOR-REST].

#### 4.1.16.2 Message Archive

When a user wants to archive a message, the client shall set the Archived flag on that message.

When a user wants to unarchive a message the client shall remove the Archived flag from that message.

For the case that permanent message storage is required due to Service Provider policy, the client is configured with the MESSAGE STORE ARCHIVE AUTH set to disabled and the client does not need to set the Archived flag.

#### 4.1.16.3 Search Operation

Search operation shall first be performed locally when local storage exists. A search may be expanded to a network based search.

#### 4.1.16.4 Message Displayed on a Client (Object Store Operation to set a flag)

When a client displays a previously unseen message for the user then it shall trigger the change of the message flag in the Common Message Store as follows:

If a message

- was originally received at a time when there the Common Message Store was permanently disabled for the client and it has not been uploaded (e.g. by user activity) or
- the message is kept only locally stored, e.g. after it was deleted from the Default folder

then the client shall manage the state only locally.

In all other cases the client shall attempt to update the message status in the Common Message Store with the procedures defined below.

If the message status is changed to seen and the client does not send a display notification then the client shall act depending on the value of the MESSAGE STORE EVENT REPORTING configuration parameter (see section A.1.3) as follows:

1. If the MESSAGE STORE EVENT REPORTING configuration parameter is disabled, the client shall set the \Seen flag for that message right after it displays the message as follows.

    - If the UID of the message for which the flag is to be set is not known because it was previously sent or received but not synchronised with the Common Message Store yet, the client shall first attempt to match it with the message in the Common Message Store as defined for synchronisation in section 4.1.16.7. If no match is found the message is handled as defined in section 4.1.16.7, otherwise it shall follow the steps below.
    - If there is no connection with the Message Store server, the client shall establish a connection to it.
    - If there is a connection to the server then the client shall use it to set the flag. If the session with the server is currently used for synchronisation as defined in section 4.1.16.7 the client shall

        - stop the synchronisation procedure at the next reasonable point (e.g. after completion of the ongoing API command or synchronisation action),
        - perform the procedure to change the flag,
        - continue the synchronisation from the point where it was stopped.

    - Then the client shall select the conversation folder where the message is stored on the server.
    - When setting the flag, the client shall follow procedures as described in section 6.6 of [CPM-MSGSTOR-REST].
    - If the Message Store server returns a success response the client shall inspect the updated flags in the response. If the result in the response is different than the local message status, then the client shall synchronise messages in the conversation folder in accordance with the definitions in section 4.1.16.7.

- If the Message Store server returns an error response the client shall synchronise messages in the conversation folder in accordance with the definitions in section 4.1.16.7.

Optimisations for setting the \Seen flags when many messages are displayed or deleted by the client in a short period of time are left to the device implementation, e.g. by keeping the Message Store connection open as long the user has the messaging application opened or until a client local inactivity timer expires.
If the connection with the server has been established by the client specifically to manage message status but not by a synchronisation trigger defined in section 4.1.16.7 the client may use this session to run a full synchronisation as defined in section 4.1.16.7.

2. If the MESSAGE STORE EVENT REPORTING configuration parameter is enabled, the client shall inform the Messaging Server as per section 4.1.14 so that the Messaging Server will set the \Seen flag.

For client fetching flags, procedures as described in section 6.6 of [CPM-MSGSTOR-REST] apply.

#### 4.1.16.5 Message Removal via Clients

When the client deletes a message from the Default folder from one device, all other clients belonging to the user shall also show that message as deleted.

When the client deletes a message which

- was originally received at a time when there the Common Message Store was permanently disabled for the client and it has not been uploaded (e.g. by user activity) or
- is kept only locally stored, e.g. after it was deleted from the Default system folder as defined in section 6.7 of [CPM-MSGSTOR-REST]

then the deletion is managed only locally.

In all other cases the client shall delete the message in the Common Message Store with the procedures defined below.

A client deletes a message from the Default folder in one of two ways, depending on the value of the MESSAGE STORE EVENT REPORTING configuration parameter (See section A.1.3):

1. If the MESSAGE STORE EVENT REPORTING configuration parameter is disabled, the client shall delete that message on the message store right after the local deletion. If the RESTful Call to the message store fails, or is not possible, the client is expected to keep a record of deleted items until the message store connection is completed and the items are successfully deleted. If there is a connection with the Message Store Server the client shall use it to delete the message.

   Optimisations for when many messages are being displayed or deleted by the client in a short period of time are to use the Bulk Delete API (section 6.4.16 of [CPM-MSGSTOR-REST]) and left to the device implementation.

2. If the MESSAGE STORE EVENT REPORTING configuration parameter is enabled, the client shall inform the Messaging Server as per section 4.1.14 so that the Messaging Server will delete the Object.

#### 4.1.16.6 Client impact of Message Removal due to Service Provider Policy

When the Message Store Server deletes (expunges) messages in the Default folder because of e.g. message expiry, the client Shall Not remove these same messages from their own device.

Clients will not receive notifications for system deleted messages over the Notification Channel, as the Message Storage server shall not create notifications for the event type of “expired object” (see Notification event types in section B.2.2).

#### 4.1.16.7 Synchronization

Client synchronization procedures are defined in section 6.3 of [CPM-MSGSTOR-REST].

NOTE: Client synchronization guidelines are built under the assumption that roaming scenarios are out of scope.

Clients using the RESTful interface shall use the notification channel to determine when to perform synchronization and this is described in section 6.2 of [CPM-MSGSTOR-REST].

When a RCS client is triggered for synchronisation it shall follow the procedures defined in section 6.7 of [CPM-MSGSTOR-REST]. Specifically it shall consider the folder structure as defined in section 4.1.7.

During synchronisation the client shall match all new sent or received messages and disposition notifications with messages of the Common Message Store as follows:

If

- a new message has been sent via SMS or MMS or
- a new message has been received via SMS or MMS or
- a delivery report has been received via SMS or MMS or
- a read report has been received or MMS or
- a flag has been changed on a message object

the client shall follow the procedures for matching of messages defined in sections 4.1.8.1 and 4.1.9.

If

- a message has been sent or
- a message has been received or
- a delivery notification has been received or
- a display notification has been received or
- a flag has been changed on a message object

via standalone messaging, chat, group chat and file transfer, the client shall follow the procedures for matching of messages as defined for these services in [CPM-MSGSTOR-REST].

For standalone messages with Pager Mode, it is possible based on Service Provider policies, to be automatically stored in the CMS as legacy SMS messages. In this case the client will not be able to correlate these messages with their copy in the Common Message Store based on the procedures for matching of messages defined in [RCS-CPM-CONVFUNC-ENDORS]. If the client is not able to correlate the standalone messages with Pager mode based on the procedures defined in [CPM-MSGSTOR-REST] it shall then attempt to correlate them based on the procedures defined for SMS messages in sections 4.1.8, 4.1.9, 4.1.10 and 4.1.11.

When a client attempts to match a new sent or received message with the Common Message Store and no match is found, then

- If the message was received via SMS or MMS and the corresponding client configuration parameter SMS MESSAGE STORE or MMS MESSAGE STORE defined in Table 85 is set to value "1", then the client shall apply the procedure defined in section 4.1.8.1.
- In all other cases the client shall keep the message locally stored. The client shall try to match the message again at least in the next synchronisation. The client shall perform message management for this message only locally.

For a new sent or received message, if a match is found, then the client shall align the local message status with the status in the Common Message Store.

The procedures for the client for setting a message seen in the Message Store Server are described in section 4.1.16.4.

The procedures for the client to delete messages in the Message Store Server are described in section 4.1.16.5.

The client shall consider the size of messages to be retrieved by examining the message meta-data in the Notification list in order to avoid delaying the fetch of the body of basic messages (chat messages, standalone messages and SMS/MMS messages). The fetch of the body of larger messages may come later in the synchronisation procedure.

If a message was originally received at a time when the Common Message Store was permanently disabled or a message is kept only locally stored, e.g. after it was deleted from the "Default" system folder as defined in section 6.7 of [CPM-MSGSTOR-REST], then the client should not consider it for synchronisation. However the user or client may consider to upload such messages to the Default folder.

Upon synchronisation, clients shall find all messages that might have been deleted on the Common Message Store by other devices.

If a message in a conversation folder of the default folder is expunged, the client shall continue to keep the message locally (as it will not receive details of expunged objects).

#### 4.1.16.8 Identification of Message Direction of Messages in a conversation

To simplify the client procedure to display messages as sent or received within a conversation an entity storing the message in the Common Message Store shall add the message transfer direction via a Direction attribute.

The Direction attribute should be stored for all types of user messages in the Common Message Store.

If the Direction attribute is present and the value is set to "Out", then the client shall display the message as sent from the own user. The client should disregard the address value in the From: header of the message. If the Direction attribute is present and the value is set to "In", then the client shall display the message as received by the own user. The client should disregard the address value in the To: header of the message.

If the Direction attribute is not present then the client should display the message direction based on a local device policy.

The Direction attribute is defined in [CPM-MSGSTOR-REST].

#### 4.1.16.9 Identification of anonymised and non-anonymised messages in a Chatbot conversation

If the 'tk' URI parameter as defined in section 2.5.4.3 is set to a value different from ‘off’ in the SIP URI associated with the folder name, then the client shall display the message as being sent in an anonymised session with that Chatbot contact.

If the 'tk' URI parameter as defined in section 2.5.4.3 is set to ‘off’ in the SIP URI associated with the folder name, then the client shall display the message as being sent in a non-anonymised session with that Chatbot contact.

The client shall handle the messages from two folders with the same Chatbot service identifier but with different values of the 'tk' URI parameter as coming from the same Chatbot.

The client shall handle the messages from a folder with an SMS number associated with the same Chatbot service identifier as coming from a non-anonymous conversation with the Chatbot.

# 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] and [PRD-RCC.15]).

### A.1.1. Configuration parameters for the management of RCS services

This RCS specification includes the following configuration parameters for the management of RCS services:

| Configuration parameter | Description | RCS usage |
|---|---|---|
| RCS DISABLED STATE | Controls the way in which an RCS client is disabled on the device. It can have following values resulting in the behaviour described in section 2.3.2.5:<br>• 0, the client is temporarily disabled<br>• -1, the client is permanently disabled<br>• -2, the client is permanently disabled, but user triggered events result in an attempt to re-enable the client<br>• -3, the client is placed in dormant state<br>If included a configuration document should not include other RCS related configuration parameters.<br>Note: if not included, the RCS client should be enabled and the document should include a valid configuration for the other RCS configuration parameters. | Optional Parameter |
| SUPPORTED RCS VERSIONS | Indicates the RCS versions supported by the configuration server. The value can be used by the client to adapt its own behaviour and its configuration requests. | Optional Parameter |
| SUPPORTED RCS PROFILE VERSIONS | Indicates the RCS profile versions supported by the configuration server. The value can be used by the client to adapt its own behaviour and its configuration requests. | Optional Parameter |

**Table 83: RCS status configuration parameters**

### A.1.2. Presence related configuration

#### *A.1.2.1. OMA Presence Provisioning parameters*

OMA Presence Client provisioning parameters are defined in [PRESENCE2MO]. Table 84 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<br>It is mandatory and becomes relevant only if DEFAULT DISCOVERY MECHANISM is set to PRESENCE |
| 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<br>It is mandatory and becomes relevant only if DEFAULT DISCOVERY MECHANISM is set to PRESENCE |
| MAX-NUMBER-OF-SUBSCRIPTIONS-IN-PRESENCE-LIST | Limits the number of back-end subscriptions allowed for a presence list. | Not used |
| SERVICE-URI-TEMPLATE | syntax of the service URI | Not Used |
| RLS-URI | SIP URI of the RLS to be used by the Watcher when subscribing to a Request-contained Presence List<br>Default if not provided:<br>*presence-rls@ims.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 | Optional parameter |

**Table 84: RCS usage of OMA presence configuration parameters**

### A.1.3. Messaging related configuration

As there are no OMA defined parameters for CPM Messaging, this RCS specification includes only RCS specific parameters. These are described in the following table:

| Configuration parameter | Description | RCS usage |
|---|---|---|
| CHAT AUTH | This parameter controls the initiation of 1-to-1 Chat sessions. When set to **0**, initiation of 1-to-1 Chat sessions is disabled. When set to **1**, initiation of 1-to-1 Chat sessions is enabled.<br>When this parameter is set to 1. GROUP CHAT AUTH parameter shall also be set to 1. | Mandatory Parameter |
| GROUP CHAT AUTH | This parameter controls the Group Chat service and receiving 1-to-1 Chat session invitations. If set to **0** the Group Chat service and receiving 1-to-1 chat invitations is disabled. When set to **1** the Group Chat service and receiving 1-to-1 chat invitations is enabled.<br>If CHAT AUTH is set to 0 (disabled), GROUP CHAT AUTH can be enabled or disabled. If CHAT AUTH is set to 1 (enabled), GROUP CHAT AUTH shall be set to 1. | Mandatory parameter |
| STANDALONE MSG AUTH | This parameter Enables/Disables the Standalone Messaging Service.<br>If set to **0** the service is disabled.<br>When set to **1** it is enabled.<br>When set to **2** it is enabled for receiving Standalone Messages but not for sending. | Mandatory Parameter |
| MAX_AD-HOC_GROUP_SIZE | Maximum number of Participants allowed for the initiation of a Group Chat. The number includes the initiator of the Group Chat. | Optional parameter<br>It is mandatory and becomes relevant only if GROUP CHAT AUTH is set to 1 or is not provided and CHAT AUTH is set to 1 |
| CONF-FCTY-URI | SIP URI used as Request URI when initiating a Group Chat.<br>Default value if not provided: chat@conf-factory.<home-network-domain-name><br>Where <home-network-domain-name> is replaced with the Home Network Domain Name used by the client (see [PRD-RCC.15] and [PRD-NG.102]) | |

| Configuration parameter | Description | RCS usage |
|---|---|---|
| EXPLODER-URI | SIP URI used as Request URI when sending 1-to-Many Standalone Messages<br>Default value if not provided: exploder@conf-factory.<home-network-domain-name><br>Where <home-network-domain-name> is replaced with the Home Network Domain Name used by the client (see [PRD-RCC.15] and [PRD-NG.102]) | |
| IM SESSION AUTO ACCEPT | This parameter controls whether the client automatically accepts incoming 1-to-1 Chat session invitations (**1**, default) or whether acceptance is sent when the user opens the Conversation thread for which the INVITE was received (**0**). | Optional parameter |
| IM SESSION AUTO ACCEPT GROUP CHAT | This parameter controls whether the client automatically accepts incoming Group Chat session invitations (**1**, default) or whether acceptance depends on a user action (**0**) for explicitly accepting the session. | Optional parameter |
| 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<br>(It is mandatory if CHAT AUTH is set to 1) |
| MAX SIZE IM | This parameter controls the maximum size in bytes of a text message that a user can enter in a 1-to-1 Chat or Group Chat session. | Optional parameter<br>(It is mandatory if CHAT AUTH is set to 1.) |
| MAX SIZE STANDALONE | This parameter controls the maximum authorised size in bytes of the content payload of a text or multimedia message without transfer encoding. | Optional parameter<br>(It is mandatory if STANDALONE MSG AUTH is set to 1.) |
| STANDALONE SWITCHOVER SIZE | The value of the configuration parameter indicates the maximum size in bytes for a Pager Mode CPM Standalone Message to switch over from the Pager Mode to the Large Message Mode.<br>If the configuration parameter is absent, then the default value 1300 shall be used. | Optional parameter |
| MESSAGE STORE URL | The URL used to access the Message Store Server<br>Default value if not provided:<br>https://msg-store.rcs.<br>mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org<br>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]). | Optional parameter |

| Configuration parameter | Description | RCS usage |
|---|---|---|
| MESSAGE STORE NOTIFICATION URL | The URL used to access the message store notification server, can be a different URL from the message store URL.<br>If not provided, the value of MESSAGE STORE URL shall be used. | |
| MESSAGE STORE USERNAME | The username for authentication with the Message Store Server via HTTP Basic or Digest authentication.<br>If the client is requested by the Message Store server for HTTP Basic or Digest authentication or the client is configured via the MESSAGE STORE AUTH parameter for initiating HTTP Basic authentication and this parameter is absent then the client shall use the value of "Realm User Name" defined in Table 2 of [PRD-RCC.15] for authentication instead. | Optional parameter |
| MESSAGE STORE PASSWORD | The password for authentication with the Message Store Server via HTTP Basic or Digest authentication. If the client is requested by the Message Store server for HTTP Basic or Digest authentication or the client is configured via the MESSAGE STORE AUTH parameter for initiating HTTP basic authentication and this parameter is absent, then the client shall use the values of "Realm User Password" defined in Table 2 of [PRD-RCC.15] for authentication instead. | Optional parameter. |
| MESSAGE STORE AUTH | This parameter controls the authentication mechanism used to access the Message Store Server.<br>**0**: Message Store server is not enabled (default)<br>**1**: Message Store server is enabled and the client shall initiate HTTP Basic authentication with user name and password from MESSAGE STORE USERNAME and MESSAGE STORE PASSWORD as described in section 4.1.5.<br>**2**: Message Store Server is enabled and the client shall perform the authentication mechanism requested by the Message Store server. | Optional parameter |
| MESSAGE STORE EVENT REPORTING | This parameter is used to inform the Message Store Client whether to directly set flags in the Message Store or whether to indicate to the Messaging Server that it should set flags in the Message Store on behalf of the client. When set to **0** the client shall set flags in the Message Store as needed via a Message Store connection (default). When set to **1**, Indicates that the client shall make use of the Event Reporting framework as described in section 4.1.14 when no Message Store connection exists so that the Messaging Server may set the flags in the Message Store on behalf of the client.<br>If not provided, the Message Store Client SHALL assume the same method as if value 0 had been specified. | Optional parameter |

| Configuration parameter | Description | RCS usage |
|---|---|---|
| MESSAGE STORE ARCHIVE AUTH | This parameter Enables/Disables the Archive service. If set to **0** the Archive service is disabled. When set to **1** it is enabled and the client may archive messages. Default value is **0**. | Optional Parameter |
| SMS MESSAGE STORE | This parameter indicates to the client whether it shall store in the RCS Default folder 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<br>(It is mandatory if MESSAGE STORE AUTH is configured) |
| MMS MESSAGE STORE | This parameter indicates to the client whether it shall store in the RCS Default folder 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<br>(It is mandatory if MESSAGE STORE AUTH is configured) |
| 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 |
| RECONNECT GUARD TIMER | This parameter is applicable when CHAT REVOKE TIMER is set to a value higher than 0. It provides the minimum time the client shall be registered in IMS prior to sending a message revocation request for a chat message when the revocation timer (i.e. CHAT REVOKE TIMER) expired.<br>Default value: 120 seconds | Optional parameter |
| CFS TRIGGER | This parameter is applicable when CHAT REVOKE TIMER is set to a value higher than 0. It controls the trigger for the client to fallback to SMS when revocation procedures apply.<br>0 (default), the client shall fall back to SMS and send the Message Revoke request right after sending the SMS<br>1, the client shall fall back to SMS right after receiving the MessageRevokeResponse request with the value of the result equal to “success” | Optional parameter |

| Configuration parameter | Description | RCS usage |
|---|---|---|
| MAX 1 TO MANY RECIPIENTS | This parameter is applicable when CHAT AUTH or STANDALONE MSG AUTH is set to 1 and it provides the maximum contacts allowed to be included in the distribution list of the 1-to-Many messaging service.<br>0 (default): the 1-to-Many Messaging service is disabled<br>1: the client can add unlimited number of recipients<br>>1: integer value that indicates the maximum total number of recipients that can be included in the distribution list | Optional parameter |
| 1 TO MANY SELECTED TECHNOLOGY | This parameter is applicable when the RCS Standalone messaging service is disabled (i.e. STANDALONE MSG AUTH is set to 0) and it controls the selected messaging technology for the 1-to-Many messaging service.<br>0 (default): SMS is selected<br>1: RCS 1-to-1 Chat service is selected | Optional parameter |
| DISPLAY NOTIFICATION SWITCH | This parameter controls whether sending of Display Notifications is enabled/disabled on the recipient's client.<br>0 (default), Enable sending Display Notifications, The user may still disable it using the Display Notification setting<br>1, Disable sending Display Notifications. The Display Notification setting is not available to the user. | Optional parameter |
| CHATBOT DIRECTORY | This parameter provides the URL from where a list of Chatbots can be retrieved as described in section 3.6.3.1. The URL shall contain the "https" scheme to enforce use of secure connections for the client's Chatbot Directory retrieval requests. | Optional parameter |
| BOTINFO FQDN ROOT | This parameter provides the root part of the FQDN to be used by the client to compose the botinfo URL as defined in section 3.6.4.1.1. | Optional parameter |
| SPECIFIC CHATBOTS LIST | This parameter provides a URL from which a list of Chatbots requiring specific management can be retrieved as described in section 3.6.3.3.<br>Default behaviour if not provided: the procedures related to the Chatbots requiring specific management are not applicable | Optional parameter |
| IDENTITY IN ENRICHED SEARCH | This parameter determines whether the i query parameter defined in section 3.6.3.1 is included in the client to Service Provider Chatbot Directory requests when the user setting to enrich the search is enabled.<br>0 (default): the i query parameters is included<br>1: the i query parameter is not included | Optional parameter |
| PRIVACY DISABLE | This parameter determines whether a user is allowed to request anonymization for Chatbot sessions.<br>When set to **0**, anonymization of Chatbot sessions is enabled (default). When set to **1**, anonymization of Chatbot sessions is disabled. | Optional parameter |

| Configuration parameter | Description | RCS usage |
|---|---|---|
| CHATBOT MSG TECH | This parameter controls the messaging technology for Chatbot messaging.<br>**0**: the Chatbot Services is disabled<br>**1** (default): the Chatbot Service is enabled with support only for 1-to-1 Chatbot sessions<br>**2**: the Chatbot Service is enabled with support for both 1-to-1 Chatbot sessions and 1-to-1 Chatbot Standalone Messages whereby for communication to a Chatbot the message technology selection described in section 3.2.1.2 applies<br>**3**: the Chatbot Service is enabled with support only for 1-to-1 Chatbot Standalone Messaging. | Optional parameter |

**Table 85: RCS Messaging related configuration parameters**

### A.1.4. File Transfer related configuration

The following configuration parameters for File Transfer, are defined in RCS:

| Configuration parameter | Description | RCS usage |
|---|---|---|
| FT AUTH | This parameter controls the Service Provider's authorisation of the File Transfer service. The following values are defined:<br>0: File Transfer is not enabled<br>1: File Transfer is enabled. | Mandatory parameter |
| FT MAX SIZE | This parameter provides a file transfer size limit in Kilobyte (KB). If a file to be transferred is bigger than FT MAX SIZE, then the client shall not initiate procedures to send the file via the File Transfer sender procedures. The configuration parameter is not applicable for the File Transfer receiver procedures.<br>If it is set to **0**, then no limit shall apply. | Mandatory parameter |
| FT WARN SIZE | This parameter provides a file transfer size limit in Kilobyte (KB). If the file size indicated in the File Transfer message body exceeds the limit of the parameter value, then the user need to accept the download prior to the invocation of the File Transfer receiver procedure.<br>If the value is set to **0**, the user shall not be warned. | Mandatory parameter |
| FT AUT ACCEPT | This parameter controls whether the default behaviour on whether a client automatically accepts incoming File Transfer invitations (**1, default**) 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. | Optional parameter |

| Configuration parameter | Description | RCS usage |
|---|---|---|
| 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 HTTP Content Server transactions.<br>Default value if not provided: https://ftcontentserver.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org<br>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]). | Optional parameter |
| FT HTTP DL URI | This parameter configures the URL of the Service Provider's local File Transfer download server. If present, the client shall apply the procedures for localisation of content URLs taken from File Transfer message bodies or other content URLs the client received via RCS messaging as defined in section 3.2.5.3.2.1.<br>If present, it is recommended to use the following FQDN value in the URL:<br>dl.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org<br>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]). | 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).<br>If not provided, the client shall use GBA authentication as described in section 3.2.5.3. | Optional parameter |
| 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).<br>If not provided, the client shall use GBA authentication as described in section 3.2.5.3. | Optional parameter |
| FT HTTP FALLBACK | This parameter provides the operator default of the technology to transfer files to contacts not supporting File Transfer via HTTP. The parameter can take the following values:<br>0: MMS (default value)<br>1: text message with a link. | Optional parameter |
| FT MAX 1 TO MANY RECIPIENTS | This parameter controls the number of recipients for the one to many sending of a File Transfer.<br>0 (default): no limitation in the number of recipients<br>positive integer value: indicates the maximum total number of recipients of a File Transfer. | Optional parameter |

**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 |
|---|---|---|
| COMPOSER AUTH | As per section 2.1.2 of [PRD-RCC.20] | As per section 2.1.2 of [PRD-RCC.20] |
| SHARED MAP AUTH | As per section 2.1.2 of [PRD-RCC.20] | As per section 2.1.2 of [PRD-RCC.20] |
| SHARED SKETCH AUTH | As per section 2.1.2 of [PRD-RCC.20] | As per section 2.1.2 of [PRD-RCC.20] |
| POST CALL AUTH | As per section 2.1.2 of [PRD-RCC.20] | As per section 2.1.2 of [PRD-RCC.20] |

**Table 87: RCS additional content sharing related configuration parameters**

### A.1.6. IMS Core / SIP related configuration

#### *A.1.6.1. IMS Provisioning Parameters*

IMS parameters shall be configured as defined in [PRD-RCC.15]. Not all parameters in the IMS Management Object defined in [3GPP TS 24.167] that is referred from [PRD-RCC.15] are relevant within the context of RCS though. Following table lists the parameters and their RCS usage:

| Configuration parameter | Description | RCS usage |
|---|---|---|
| ConRef | Represents a network access point object | Mandatory parameter<br>Not used for RCS<br>Provided with dummy value if RCS is only user of MO: *dummy.apn*<br>Selection of the APN for RCS depends on other parameters as defined in section 2.8.1.4 |

| Configuration parameter | Description | RCS usage |
|---|---|---|
| PDP_ContextOperPref | Indicates an operator's preference to have a dedicated Packet Data Protocol (PDP) context for SIP signalling. | Mandatory parameter<br>Not used for RCS, always set to 0 (no preference) if RCS is only user of MO. |
| P-CSCF_Address | an FQDN or an IPv4 address to an IPv4 P-CSCF | Optional parameter<br>Not provided if RCS is only user of MO. The P-CSCF address for RCS is provided in LBO_P-CSCF_Address |
| Timer_T1 | Defines the SIP timer T1 – the RTT estimate | Mandatory parameter |
| Timer_T2 | Defines the SIP timer T2 – the maximum retransmit interval for non-INVITE requests and INVITE responses. | Mandatory parameter |
| Timer_T4 | Defines the SIP timer T4 – the maximum duration a message will remain in the network. | Mandatory parameter |
| Private_user_identity | Represents the private user identity<br>Note: It is recommended to set it to the same value as for the configuration parameter Realm User Name defined in section 2.2.1.2 of [PRD-RCC.15]. | Mandatory parameter<br>Used as defined in [PRD-RCC.15] |
| Public_user_identity | Represents a public user identity. | Mandatory parameter<br>Used as defined in [PRD-RCC.15] and if applicable [PRD-NG.102] |

| Configuration parameter | Description | RCS usage |
|---|---|---|
| Home_network_domain_name | Indicates the operator's home network domain. | Mandatory parameter<br>Used as defined in [PRD-RCC.15] and if applicable [PRD-NG.102]<br>Recommended to use ims.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]). |
| ICSI_List | Used to allow a reference to a list of IMS communication service identifiers that are supported by a subscriber's network for that subscriber. | Mandatory Tree<br>Tree is not used for RCS, but mandatory to be provided. No leafs shall be provided if RCS is only user of MO. |
| LBO_P-CSCF_Address | Provides a reference to a list of P-CSCFs | Optional tree<br>Used as defined in [PRD-RCC.15] and if applicable [PRD-NG.102] |
| Address (LBO_P-CSCF_Address) | Defines the FQDN, the IPv4 address or the IPv6 address of a P-CSCF | Mandatory parameter |
| AddressType (LBO_P-CSCF_Address) | Defines the type of address stored in the Address leaf node | Mandatory parameter |

| Configuration parameter | Description | RCS usage |
|---|---|---|
| Resource_Allocation_Mode | Indicates whether UE initiates resource allocation for the media controlled by IM CN subsystem for all IMS sessions not covered by any "ICSI Resource Allocation Mode", when both UE and network can initiate resource allocation | Optional parameter<br>Parameter not provided if RCS is only user of MO. |
| Voice_Domain_Preference_E_UTRAN | Indicates network operator's preference for selection of the domain to be used for voice communication services by the UE. | Optional parameter<br>Parameter not provided if RCS is only user of MO. |
| SMS_Over_IP_Networks_Indication | Indicates network operator's preference for selection of the domain to be used for short message service (SMS) originated by the UE. | Optional parameter<br>Parameter not provided if RCS is only user of MO. |
| Keep_Alive_Enabled | Indicates whether the UE sends keep alives | Mandatory parameter |
| Voice_Domain_Preference_UTRAN | Indicates network operator's preference for selection of the domain to be used for voice communication services by the UE. | Optional parameter<br>Parameter not provided if RCS is only user of MO. |
| /<X>/Mobility_Management_IMS_Voice_Termination | Indicates whether the UE mobility management performs additional procedures to support terminating access domain selection by the network. | Optional parameter<br>Parameter not provided if RCS is only user of MO. |
| RegRetryBaseTime | Represents the value of the base-time parameter | Optional parameter |
| RegRetryMaxTime | Represents the value of the max-time parameter | Optional parameter |
| PhoneContext_List | Used to allow a reference to a list of phone-context parameter values for other local numbers, than geo-local or home-local numbers | Optional tree<br>Tree is not provided for RCS because only home-local and geo-local numbers are used (see section 2.5) |
| SS_domain_setting | Indicates the network operator's preference for the selection of the domain used by the UE when performing supplementary services (SS) setting control for voice services. | Optional parameter<br>Parameter not provided if RCS is only user of MO. |

| Configuration parameter | Description | RCS usage |
|---|---|---|
| PS_domain_IMS_SS_control_preference | Provides a means to define the method for which Supplementary Services are controlled by the UE when SS setting control is to be invoked over the PS domain. | Optional parameter<br>Parameter not provided if RCS is only user of MO. |

**Table 88: Usage of [3GPP TS 24.167] MO parameters for RCS**

#### *A.1.6.2. RCS Specific Provisioning parameters*

This RCS specification includes the following additional IMS Core/SIP related configuration parameters:

| Configuration parameter | Description | RCS usage |
|---|---|---|
| RCS VOLTE SINGLE REGISTRATION | This parameter defines the behaviour regarding the registration for RCS services for devices supporting the IMS well-known APN.<br>0, the device *Shall* follow a dual registration approach (transition solution)<br>1 (default if not provided), the device *Shall* follow a single registration (target solution).<br>2, the device Shall follow a single registration for all RCS services (i.e. including Multimedia Telephony and SMSoIP) when in the home network, and Shall follow a dual registration approach when roaming (transition solution).<br>The parameter is also used to control the APN selection on devices that are not enabled for VoLTE/VoWiFi as described in section 2.8.1.4.<br>Devices not supporting the IMS well-known APN shall ignore the configuration parameter. | Optional Parameter |

**Table 89: RCS additional IMS Core/SIP related configuration parameters**

### A.1.7. Geolocation related configuration

This RCS specification includes the following 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 |

**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. Capability discovery related configuration

This RCS specification includes the following RCS Specific configuration parameters related to the capability discovery:

| Configuration parameter | Description | RCS usage |
|---|---|---|
| DISABLE INITIAL ADDRESS BOOK SCAN | This parameter controls whether the device/client performs a capability check for all contacts in the address book when it is first started.<br>When set to **0** (Default value), the device/client shall perform the scan<br>When set to **1**, the device shall skip the scan and only perform capability exchange requests based on the other triggers defined in section 2.6. | Optional parameter |
| CAPABILITY INFO EXPIRY | When using the capability discovery mechanism and with the aim of minimising the traffic, an expiry time is set in the capability information for real-time and non-real-time communication services fetched using SIP OPTIONS or Presence.<br>When capability information was obtained more recently than the value configured for this parameter, it shall be considered as being still valid.<br>When set to 0, the cached capabilities shall be considered to never expire and shall be invalidated only by a conclusive capability and service availability exchange indicating that they are no longer valid<br>Default value:2592000 (30 days) | Optional parameter |
| SERVICE AVAILABILITY INFO EXPIRY | This parameter controls expiration of cached service availability information of contacts.<br>Its value shall indicate the validity time of cached availability information in seconds.<br>Default value: 60 | Optional Parameter |
| CAPABILITY DISCOVERY MECHANISM | This parameter allows selecting the default capability and new user discovery mechanism.<br>If set to OPTIONS (**0**), the default mechanism employed for capability discovery and new users will be OPTIONS.<br>If set to PRESENCE (**1**), the mechanism employed for capability discovery and new users will relay presence-based information.<br>If not provided or set to OFF (**2**) the capability discovery mechanism is disabled. | Optional parameter |

| Configuration parameter | Description | RCS usage |
|---|---|---|
| CAPABILITY DISCOVERY ALLOWED PREFIXES | The configuration parameter provides prefixes or rules to identify phone numbers contained in the address book or entered by the user which shall be considered for capability and new user discovery.<br>If the configuration parameter is absent, all phone numbers shall be considered for capability and new user discovery.<br>If a number shall be considered for capability and new user discovery, then the client shall invoke the procedures for capability and new user discovery in accordance with the definitions in section 2.6.<br>If a number shall not be considered for capability and new user discovery, the client shall not invoke the procedures for capability and new user discovery defined in section 2.6 and shall consider the phone number as not RCS capable. If capability and new user discovery through SIP OPTIONS is applied, the client shall process received discovery request in accordance with the definitions in section 2.6.1.1.<br>The service provider should take the subscriber's HPLMN numbering scheme into account when defining the value of the configuration parameter. | Optional parameter |
| NON RCS CAPABILITY INFO EXPIRY | This parameter allows to better control the amount of capability query sent to non RCS contacts.<br>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.<br>Default value: 2592000 (30 days) | Optional parameter |

**Table 91: RCS additional capability discovery related configuration parameters**

### A.1.10. APN configuration

This RCS specification includes the following RCS Specific configuration parameters targeting APN configuration (see sections 2.8.1.4 and 2.12):

| Configuration parameter | Description | RCS usage |
|---|---|---|
| NO MSRP SUPPORT | This parameter lists networks outside of the home network knows not supporting MSRP.<br>If the value of the configuration parameter RCS VOLTE SINGLE REGISTRATION defined in section A.1.7.2. is set to "1", then the client the client shall use the HOS APN for the registration of RCS services if connected to the cellular access of a network outside of the home network, if the network is listed in the configuration parameter.<br>If not instantiated, the device shall behave as if it was an empty list.<br>For all other clients and devices the configuration parameter is not applicable. | Optional Parameter |

**Table 92: RCS roaming configuration parameters**

### A.1.11. 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 RCS IP VOICE CALL | This parameter allows to enable or to disable the RCS IP Voice Call Service on non-VoLTE/VoWiFi enabled primary devices for use over non-cellular access and on secondary devices regardless of the access. | Mandatory Parameter |
| PROVIDE RCS IP VIDEO CALL | This parameter allows to enable or to disable the RCS IP Video Call Service on devices depending on network connectivity (only non-3GPP/non-3GPP2 networks, also on LTE, etc.). | Mandatory Parameter |

**Table 93: RCS IP Voice and Video Call configuration parameters**

### A.1.12. 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.1.13. Plug-in configuration parameters

The RCS specification includes the following additional Plug-in related configuration parameters:

| Configuration parameter | Description | RCS usage |
|---|---|---|
| CATALOG URI | The URI used to construct the Catalog retrieval URL to access the Plug-in Info server for initially retrieving or refreshing the Catalog. The parameter is provided when the Service Provider wants to offer the list of Plug-ins to their users. If not provided, Plug-ins are not available to the user. The URI shall contain the "https" schema to enforce use of secure connections for the client's Catalog retrieval requests. | Optional parameter |

**Table 94: RCS extensions configuration parameters**

### A.1.14. Data Off

#### *A.1.14.1. RCS Configuration Parameters*

The RCS specification includes following configuration parameters controlling the behaviour of the respective services when connected over cellular networks using a primary device that is not using the internet APN for RCS (see section 2.8.1.4) and data is switched off by the user:

| Configuration parameter | Description | RCS usage |
|---|---|---|
| RCS MESSAGING DATA OFF | This parameter indicates whether the 1-to-1 and Group Chat, Standalone Messaging and Geolocation PUSH services *Should* remain available in case the cellular data switch is switched off (either toggled manually by the user or automatically during roaming).<br>When set to 0 the Chat, Standalone Messaging and Geolocation PUSH services are not cellular data off exempt services on cellular networks when cellular data is switched off.<br>When set to 1 the Chat, Standalone Messaging and Geolocation PUSH services are cellular data off exempt services on cellular networks when cellular Data is switched off.<br>When set to 2 the Chat, Standalone Messaging and Geolocation PUSH services are cellular data off exempt services on cellular networks when cellular Data is switched off and the device is attached to the VPLMN. | Mandatory Parameter |

| Configuration parameter | Description | RCS usage |
|---|---|---|
| FILE TRANSFER DATA OFF | This parameter indicates whether the File Transfer service *Should* remain available in case the cellular data switch is switched off (either toggled manually by the user or automatically during roaming).<br>When set to 0 the File Transfer Service is not a cellular data off exempt service on cellular networks when cellular data is switched off.<br>When set to 1 the File Transfer service is a cellular data off exempt service on cellular networks when cellular Data is switched off.<br>When set to 2 the File Transfer service is a cellular data off exempt service on cellular networks when cellular Data is switched off and the device is attached to the VPLMN.<br>When File Transfer is disabled, then the client shall behave as defined in section 3.2.5.2 for the case where File Transfer is not authorised via client configuration. | Mandatory Parameter |
| MMS DATA OFF | This parameter indicates whether MMS should remain available in case the cellular data switch is switched off (either toggled manually by the user or automatically during roaming).<br>When set to **0** MMS is not a cellular data off exempt service when cellular data is switched off.<br>When set to **1** (default value) MMS a cellular data off exempt service when cellular Data is switched off.<br>When set to **2** MMS is a cellular data off exempt service when cellular data is switched off and the device is attached to the HPLMN, otherwise MMS is disabled.<br>NOTE: the device's settings to enable/disable automatic download of received MMS messages remain applicable. | Optional Parameter |
| CONTENT SHARE DATA OFF | This parameter indicates whether the Shared Map and Shared Sketch services should remain available in case cellular data is switched off (either toggled manually by the user or automatically during roaming).<br>When set to **0** the Shared Map and Shared Sketch services are not a cellular data off exempt services on cellular networks when cellular data is switched off.<br>When set to **1** the Shared Map and Shared Sketch services are cellular data off exempt services on cellular networks when cellular data is switched off.<br>When set to **2** the Shared Map and Shared Sketch services are cellular data off exempt services on cellular networks when cellular data is switched off and the device is attached to the VPLMN. | Optional Parameter<br>It becomes mandatory if SHARED MAP AUTH and/or SHARED SKETCH AUTH is set to 1 (see section 2.1.2 of [PRD-RCC.20]) |

| Configuration parameter | Description | RCS usage |
|---|---|---|
| PRE AND POST CALL DATA OFF | This parameter indicates whether the Call Composer and Post Call services should remain available in case cellular data is switched off (either toggled manually by the user or automatically during roaming).<br>When set to **0** the Call Composer and Post-call services are not a cellular data off exempt services on cellular networks when cellular data is switched off.<br>When set to **1** the Call Composer and Post-call services are cellular data off exempt services on cellular networks when cellular data is switched off.<br>When set to **2** the Call Composer and Post-call services are cellular data off exempt services on cellular networks when cellular data is switched off and the device is attached to the VPLMN. | Optional Parameter<br>It becomes mandatory if CALL COMPOSER AUTH and/or POST CALL AUTH is set to a value other than 0 (see section 2.1.2 of [PRD-RCC.20]) |
| SYNC DATA OFF | This parameter indicates whether the synchronization with the Common Message Store should remain available in case the cellular data is switched off (either toggled manually by the user or automatically during roaming).<br>When set to **0** the synchronisation with the Common Message Store is not a cellular data off exempt service when cellular data is switched off.<br>When set to **1** synchronisation with the Common Message Store is a cellular data off exempt service when cellular data is switched off.<br>When set to **2** the synchronisation with the Common Message Store is a cellular data off exempt service when cellular data is switched off and the device is attached to the VPLMN.<br>If the synchronisation with the Common Message Store is not a cellular data off exempt service, the client shall not invoke the synchronisation with the Common Message Store, shall not set message flags via the Message Store and the event notification framework and shall not store SMS and MMS messages. Once synchronisation is enabled again, the client shall invoke the procedures for the synchronisation with the Common Message Store for the missed events. | Optional Parameter<br>It becomes mandatory if MESSAGE STORE AUTH is present (see A.1.3) |

**Table 95: RCS Data Off Configuration Parameters**

NOTE1: No parameter is provided for RCS IP Voice Calls because for primary devices they are only available on the Wi-Fi bearer.

NOTE2: These parameters only affect behaviour on cellular networks. Services that can be offered over non-cellular networks remain available over such networks irrespective of the setting of the cellular data switch. These parameters also affect services when RCS uses the HOS APN.

#### *A.1.14.2. Non-Access Stratum Configuration Parameters*

RCS uses the configuration parameters of the Non-Access Stratum Management Object defined in [3GPP TS 24.368] to control the behaviour of the respective services when connected over cellular networks using a primary device that is not using the internet APN for RCS (see section 2.8.1.4) and data is switched off by the user. The following parameters of the non-access stratum management object are used.

| Configuration parameter | Description | RCS usage |
|---|---|---|
| Device Management over PS data off exemption | This parameter indicates whether discovery of plug-ins according to section 3.2.8 and client configuration for chatbots according to section 3.6.3.3 are cellular data off exempt services.<br>The following values are defined:<br>0: the device management services are not defined as a cellular data off exempted services.<br>1: the device management services are defined as a cellular data off exempted services (default value). | Optional Parameter |
| Device Management over PS data off roaming exemption | This parameter indicates whether discovery of plug-ins according to section 3.2.8 and client configuration for chatbots according to section 3.6.3.3 are cellular data off exempt services when roaming.<br>The following values are defined:<br>0: the device management services are not defined as a cellular data off exempted services when roaming.<br>1: the device management services are defined as a cellular data off exempted services when roaming (default value). | Optional Parameter |

**Table 96: Non-Access Stratum Configuration Parameters**

## A.2. Provisioning Document of the RCS Management tree

The RCS Management tree data is conveyed between the configuration server and the client by use of a configuration XML document as defined in section 4 of [PRD-RCC.14].

This section defines the application characteristic for the management of RCS configuration parameters.

### A.2.1. Application characteristic type for the RCS Management tree

The following parameters and values are defined for the application characteristic type for the use in the RCS Management tree.

The AppId parameter is used to uniquely identify the RCS Management tree in a configuration XML document.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | chr | Get |

**Table 97: Application characteristic type parameter AppID**

- Value: ap2002

- Post-reconfiguration actions: n/a
- Associated HTTP XML parameter: “AppID”

The To-Appref parameter provides the link between the RCS Management tree application characteristic with an instance of the IMS MO identified by the APPREF parameter defined in [PRD-RCC.15].

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | chr | Get |

**Table 98: Application characteristic type parameter To-Appref**

- Value: string containing the reference identity of the linked IMS MO instance
- Post-reconfiguration actions: If the value changes, the client shall re-register or de-register in IMS to disable RCS services and RCS services and register or re-register in IMS for RCS services using the new IMS core network configuration.
- Associated HTTP XML parameter: “To-Appref”

```
<characteristic type=”APPLICATION”>
        <parm name=”AppID” value=”ap2002”/>
        <parm name=”To-Appref” value=”X”/>
        <!-- other characteristics of the RCS configuration are embedded here -->
</characteristic>
```

**Table 99: Application type XML structure**

### A.2.2. 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:

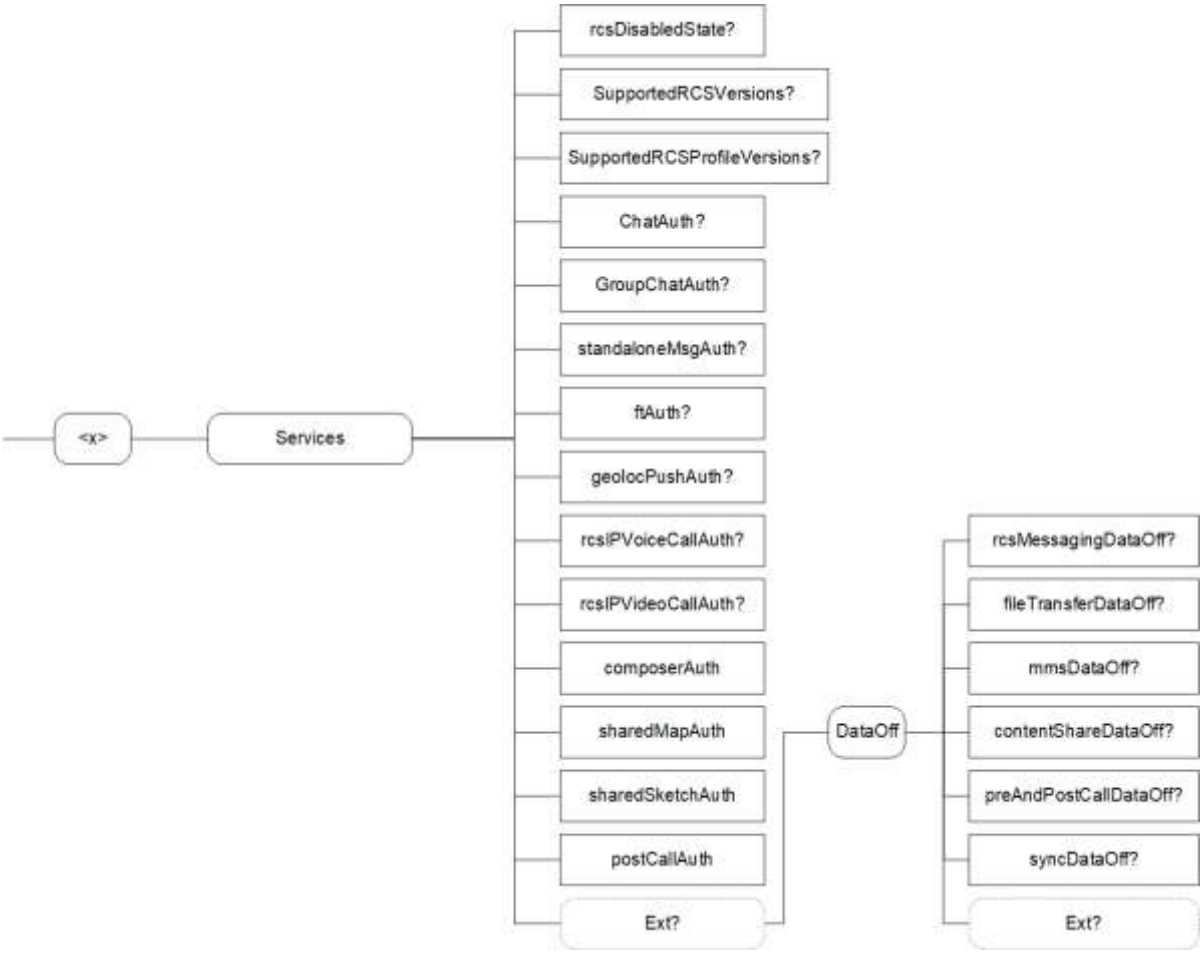



**Figure 19: RCS additions, Services sub tree**

The associated HTTP configuration XML structure is presented in the table below:

```
<characteristic type="SERVICES">
        <parm name="SupportedRCSVersions" value="X"/>
        <parm name="SupportedRCSProfileVersions" value="X"/>
        <parm name="ChatAuth" value="X"/>
        <parm name="GroupChatAuth" value="X"/>
        <parm name="ftAuth" value="X"/>
        <parm name="standaloneMsgAuth" value="X"/>
        <parm name="geolocPushAuth" value="X"/>
        <parm name="rcsIPVoiceCallAuth" value="X"/>
        <parm name="rcsIPVideoCallAuth" value="X"/>
        <parm name="composerAuth" value="X"/>
        <parm name="sharedMapAuth" value="X"/>
        <parm name="sharedSketchAuth" value="X"/>
        <parm name="postCallAuth" value="X"/>
        <characteristic type="Ext">
              <characteristic type="DataOff">
                    <parm name="rcsMessagingDataOff" value="X"/>
                    <parm name="fileTransferDataOff" value="X"/>
                    <parm name="mmsDataOff" value="X"/>
                    <parm name="contentShareDataOff" value="X"/>
                    <parm name="preAndPostCallDataOff" value="X"/>
                    <parm name="syncDataOff" value="X"/>
                    <characteristic type="Ext"/>
              </characteristic>
        </characteristic>
</characteristic>
```

**Table 100 : Services MO sub tree associated HTTP configuration XML structure**

Note: rcsDisabledState is not included in Table 100 since the presence of that parameter would invalidate most of the other parameters included.

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 101: Services MO sub tree addition services node**

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-services:9.0*
- Associated HTTP XML characteristic type: “SERVICES”

Node: /<x>/Services/rcsDisabledState

Leaf node that controls the state of the RCS services on the device

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | int | Get, Replace |

**Table 102: Services MO sub tree addition parameters (rcsDisabledState)**

- Values:
  0, the client is temporarily disabled

-1, the client is permanently disabled
-2, the client is permanently disabled, but user triggered events result in an attempt to re-enable the client
-3, the client is placed in dormant state

- Post-reconfiguration actions: The client shall change the state of the relevant RCS services as described in section 2.3.2.5.
- Associated HTTP XML parameter: “rcsDisabledState”

NOTE: Given that due to the definition in section A.1.1, this parameter and the other parameters in this management object are mutually exclusive it is not included in Table 100 and Table 188.

Node: /<x>/Services/SupportedRCSVersions

Leaf node that indicates the RCS versions supported by the configuration server

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | chr | Get, Replace |

**Table 103: Services MO sub tree addition parameters (SupportedRCSVersions)**

- Values:
  Comma separated list of RCS versions supported by the configuration server. The RCS versions are defined in the rcs_version parameter for the client configuration request in the associated version of this document.
  Example:
  5.1B,6.0,7.0,8.0
- Post-reconfiguration actions: No action at the time of re-configuration.
- Associated HTTP XML parameter ID: “SupportedRCSVersions”

Node: /<x>/Services/SupportedRCSProfileVersions

Leaf node that indicates the RCS profile versions supported by the configuration server

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | bool | Get, Replace |

**Table 104: Services MO sub tree addition parameters (SupportedRCSProfileVersions)**

- Values:
  Comma separated list of RCS profile versions supported by the configuration server. The RCS profile versions are defined in the rcs_profile parameter for the client configuration request in [PRD-RCC.71] .
  Example:
  foo,bar
- Post-reconfiguration actions: No action at the time of re-configuration.
- Associated HTTP XML parameter ID: “SupportedRCSProfileVersions”

Node: /<x>/Services/ChatAuth

Leaf node that represents the authorisation for the user to send chat messages

The node shall be instantiated if the rcsDisabledState node is not provided.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | bool | Get, Replace |

**Table 105: Services MO sub tree addition parameters (ChatAuth)**

- Values: 0, 1
  0- Indicates that initiating 1-to-1 chat session is disabled
  1- Indicates that initiating 1-to-1 chat sessions is enabled
- Post-reconfiguration actions: No action at the time of re-configuration.
- Associated HTTP XML parameter ID: “ChatAuth”

Node: /<x>/Services/GroupChatAuth

Leaf node that represents the authorisation for the user to receive chat messages and use the group chat service

The node shall be instantiated if the rcsDisabledState node is not provided.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | bool | Get, Replace |

**Table 106: Services MO sub tree addition parameters (GroupChatAuth)**

- Values: 0, 1
  0- Indicates that Group Chat service and receiving 1-to-1 chat session invitations is disabled
  1- Indicates that Group Chat and receiving 1-to-1 chat session invitations service is enabled
- Post-reconfiguration actions: If the value transits from "0" to "1" then the client shall register or re-register in IMS to add the feature tags for Chat defined in section 2.4.4, if conditions allow. If the value of transits from "1" to "0" and if the client is registered in IMS for Chat and Group Chat as defined in section 2.4.4, then the client shall initiate an IMS de-registration or re-registration to remove the feature tags for Chat defined in section 2.4.4.
- Associated HTTP XML parameter ID: “GroupChatAuth”

Node: /<x>/Services/ftAuth

Leaf node that represent the authorisation for user to use the File Transfer

The node shall be instantiated if the rcsDisabledState node is not provided.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | bool | Get, Replace |

**Table 107: Services MO sub tree addition parameters (ftAuth)**

- Values: 0, 1
  0- Indicates that File Transfer is disabled
  1- Indicates that File Transfer is enabled

- Post-reconfiguration actions: If the value of configuration parameter transits from 0 to 1 and the client is registered in IMS, then the client shall re-register to add the media feature tag defined for File Transfer. If the value of configuration parameter transits from 1 to 0 and the client is registered in IMS, then the client shall re-register to remove the media feature tag defined for File Transfer.
- Associated HTTP XML parameter ID: “ftAuth”

Node: /<x>/Services/standaloneMsgAuth

Leaf node that represents the authorisation for user to use the standalone messaging service

The node shall be instantiated if the rcsDisabledState node is not provided.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | int | Get, Replace |

**Table 108: Services MO sub tree addition parameters (standaloneMsgAuth)**

- Values: 0, 1, 2
  0- The standalone messaging service is not provided. SMS and MMS is used instead
  1- Sending and receiving of messages via the standalone messaging service is provided and uses CPM as specified in [RCS-CPM-CONVFUNC-ENDORS].
  2- Receiving of messages via the standalone messaging service is provided and uses CPM as specified in [RCS-CPM-CONVFUNC-ENDORS]. SMS and MMS are still used for sending.
- Post-reconfiguration actions:
  - If the value of the configuration parameter transits from value 0 to value 1 or 2, the client may wait till the next scheduled refresh re-REGISTER request or may issue re-REGISTER request immediately.
  - If the value of the configuration parameter transits from value 1 or 2 to value 0, the client may wait till the next scheduled refresh re-REGISTER request or may issue re-REGISTER request immediately.
  - If the value of the configuration parameter transits from value 1 to value 2, the client shall stop sending messages via the standalone messaging service.
  - If the value of the configuration parameter transits from value 2 to value 1, the client shall be able to send messages via the standalone messaging service.
- Associated HTTP XML parameter ID: “standaloneMsgAuth”

Node: /<x>/Services/geolocPushAuth

Leaf node that represents the authorisation for the user to use the Geolocation PUSH service

The node shall be instantiated if the rcsDisabledState node is not provided.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | bool | Get, Replace |

**Table 109: 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.
- Associated HTTP XML parameter ID: “geolocPushAuth”

Node: /<x>/Services/rcsIPVoiceCallAuth

Leaf node that represents the authorisation for user to use RCS IP Voice Call service on a secondary device

The node shall be instantiated if the rcsDisabledState node is not provided.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | int | Get, Replace |

**Table 110: Services MO sub tree addition parameters (rcsIPVoiceCallAuth)**

- Values: 0 or 1:
- 0- Indicates that the RCS IP Voice Call service is disabled on secondary devices
  1- Indicates that the RCS IP Voice Call service is enabled on secondary devices
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration as described in section 2.3
- Associated HTTP XML parameter ID: “rcsIPVoiceCallAuth”

Node: /<x>/Services/rcsIPVideoCallAuth

Leaf node that represents the authorisation for user to use the RCS IP Video Call service

The node shall be instantiated if the rcsDisabledState node is not provided.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | int | Get, Replace |

**Table 111: 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 112: 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 Wi-Fi, 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 as described in section 2.3.
- Associated HTTP XML parameter ID: “rcsIPVideoCallAuth”

Node: /<x>/Services/composerAuth

As per 2.1.2.1 of [PRD-RCC.20]

Node: /<x>/Services/sharedMapAuth

As per 2.1.2.1 of [PRD-RCC.20]

Node: /<x>/Services/sharedSketchAuth

As per 2.1.2.1 of [PRD-RCC.20]

Node: /<x>/Services/postCallAuth

As per 2.1.2.1 of [PRD-RCC.20]

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 113: Services MO sub tree addition Service Provider Extension Node**

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-services:9.0:Ext*
- Post-reconfiguration actions: The client should be reset and should perform the complete first-time registration procedure following a reconfiguration as described in section 2.3.
- Associated HTTP XML characteristic type: “EXT”

Node: /<x>/Services/Ext/DataOff

Under this interior node where the specific RCS parameters are placed that relate to the services behaviour on cellular networks when the cellular data switch is switched off.

It shall be instantiated for primary devices where it is required to be supported.

NOTE: This tree is included as part of the ext tree rather than sitting directly under Services for historic reasons

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | Node | Get |

**Table 114: Data Off Services Extension MO sub tree addition node**

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-services:9.0:Ext:DataOff*
- Associated HTTP XML characteristic type: “DataOff”

Node: /<x>/Services/Ext/DataOff/rcsMessagingDataOff

Controls the Chat, Standalone Messaging and Geolocation PUSH service behaviour when the cellular data switch is switched off.

The parameter is only applicable in case the Chat or Standalone services are supported. It will not be instantiated otherwise.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | Int | Get, Replace |

**Table 115: Data Off Services Extension MO sub tree addition parameters (rcsMessagingDataOff)**

- Values:
  0: the Chat, Standalone Messaging and Geolocation PUSH services are not cellular data off exempt services
  1: the Chat, Standalone Messaging and Geolocation PUSH services are cellular data off exempt services
  2: the Chat, Standalone Messaging and Geolocation PUSH services are cellular data off exempt services if the device is attached to the VPLMN.

- Post-reconfiguration actions: If the value of the configuration parameter transits from 0 to 1 or from 0 to 2 while the device is connected to the HPLMN and at least one RCS messaging service is authorised the client shall (re-)register in IMS to add the relevant media feature tags for Chat, File Transfer, Standalone Messaging and Geolocation PUSH services according to the authorisation of these services.
  If the value of the configuration parameter transits from 1 to 0 or from 2 to 0 while the device is connected to a cellular access network other than the HPLMN and at least one of the RCS messaging services is registered in IMS, the client shall de- or re-register with in IMS to remove the media feature tags for Chat, File Transfer, Standalone Messaging or Geolocation PUSH services if these have been registered.
- Associated HTTP XML characteristic type: “rcsMessagingDataOff”

Node: /<x>/Services/Ext/DataOff/fileTransferDataOff

Controls the File Transfer service behaviour when the cellular data switch is switched off.

The parameter is only applicable in case the File Transfer service is supported. It will not be instantiated otherwise.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | Int | Get, Replace |

**Table 116: Data Off Services Extension MO sub tree addition parameters (fileTransferDataOff)**

- Values:
  0: the File Transfer service is not a cellular data off exempt service
  1: the File Transfer service is a cellular data off exempt service
  2: the File Transfer service is a cellular data off exempt service if the device is attached to the VPLMN.
- Post-reconfiguration actions: If the value of the configuration parameter transits from 0 to 1 or from 0 to 2 while the device is connected to the HPLMN and the File Transfer service is authorised, the client shall (re-)register in IMS to add the media feature tag for File Transfer via HTTP.
  If the value of the configuration parameter transits from 1 to 0 or from 2 to 0 while the device is connected to a cellular access network other than the HPLMN and File Transfer via HTTP is registered in IMS, the client shall de- or re-register in IMS to remove the media feature tag for File Transfer via HTTP.
- Associated HTTP XML characteristic type: “fileTransferDataOff”

Node: /<x>/Services/Ext/DataOff/mmsDataOff

Controls the MMS behaviour when the cellular data switch is switched off.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | Int | Get, Replace |

**Table 117: Data Off Services Extension MO sub tree addition parameters (mmsDataOff)**

- Values:
  0: the MMS is not a cellular data off exempt service.
  1: the MMS is a cellular data off exempt service
  2: the MMS is a cellular data off exempt service if the client is attached to the VPLMN.
- Post-reconfiguration actions: no specific actions.
- Associated HTTP XML characteristic type: “mmsDataOff”

Node: /<x>/Services/Ext/DataOff/contentShareDataOff

Controls the Shared Map and Shared Sketch service behaviour when the cellular data switch is switched off.

The parameter is only applicable in case the Shared Map and Shared Sketch services are enabled. It will not be instantiated otherwise.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | Int | Get, Replace |

**Table 118: Data Off Services Extension MO sub tree addition parameters (contentShareDataOff)**

- Values:
  0: the Shared Map and Shared Sketch services are not cellular data off exempt services
  1: the Shared Map and Shared Sketch services are cellular data off exempt services
  2: the Shared Map and Shared Sketch services are cellular data off exempt services when the device is attached to the VPLMN.
- Post-reconfiguration actions: If the value of the configuration parameter transits from 0 to 1 or from 0 to 2 while the device is connected to the HPLMN and at least one of Shared Map or Shared Sketch is authorised the client shall (re-)register in IMS to add the relevant media feature tags according to the authorisation of these services. If the value of the configuration parameter transits from 1 to 0 or from 2 to 0 while the device is connected to a cellular access network other than the HPLMN and at least one of the Shared Map or Shared Sketch service is registered in IMS, the client shall de-or re-register with in IMS to remove the media feature tags for the service being disabled by the configuration parameter.
- Associated HTTP XML characteristic type: “contentShareDataOff”

Node: /<x>/Services/Ext/DataOff/preAndPostCallDataOff

Controls the Call Composer and Post-call service behaviour when the cellular is switched off.

The parameter is only applicable in case the Call Composer and/or Post-call services are supported. It will not be instantiated otherwise.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | Int | Get, Replace |

**Table 119: Data Off Services Extension MO sub tree addition parameters (preAndPostCallDataOff)**

- Values:
  0: the Call Composer and Post-call services are not cellular data off exempt services

  1: the Call Composer and Post-call services are cellular data off exempt services
  2: the Call Composer and Post-call services are cellular data off exempt services ifthe device is attached to the VPLMN.
- Post-reconfiguration actions: If the value of the configuration parameter transits from 0 to 1 or from 0 to 2 while the device is connected to the HPLMN and at least one of Call Composer and Post-call is authorised the client shall (re-)register in IMS to add the relevant media feature tags according to the authorisation of these services.
  If the value of the configuration parameter transits from 1 to 0 or from 2 to 0 while the device is connected to a cellular access network other than the HPLMN and at least one of Call Composer and Post-call service is registered in IMS, the client shall de- or re-register with in IMS to remove the media feature tags for the service being disabled by the configuration parameter.
- Associated HTTP XML characteristic type: “preAndPostCallDataOff”

Node: /<x>/Services/Ext/DataOff/syncDataOff

Controls the behaviour for synchronisation with the Common Message when the cellular data is switched off.

The parameter is only applicable in case the Common Message Store is supported. It will not be instantiated otherwise.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | Int | Get, Replace |

**Table 120: Data Off Services Extension MO sub tree addition parameters (syncDataOff)**

- Values:
  0: the synchronisation with the Common Message Store is not a cellular data off exempt service
  1: the synchronisation with the Common Message Store is a cellular data off exempt service
  2: the synchronisation with the Common Message Store is a cellular data off exempt service if the device is attached to the VPLMN.
- Post-reconfiguration actions: If the value of the configuration parameter transits from 1 to 0 or from 2 to 0 while the device is connected to a cellular access network other than the HPLMN the client should trigger a data connection triggered synchronization with the Common Message Store.
- Associated HTTP XML characteristic type: “syncDataOff”

### A.2.3. Presence sub tree

The HTTP configuration XML structure associated with the Presence parameters from the Presence MO defined in [PRESENCE2MO] is presented in the table below

```
<characteristic type="PRESENCE">
        <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 121: Presence sub tree associated HTTP configuration XML structure**

### A.2.4. Messaging sub tree additions

RCS includes the following additions as a new configuration sub tree, the Messaging MO subtree:

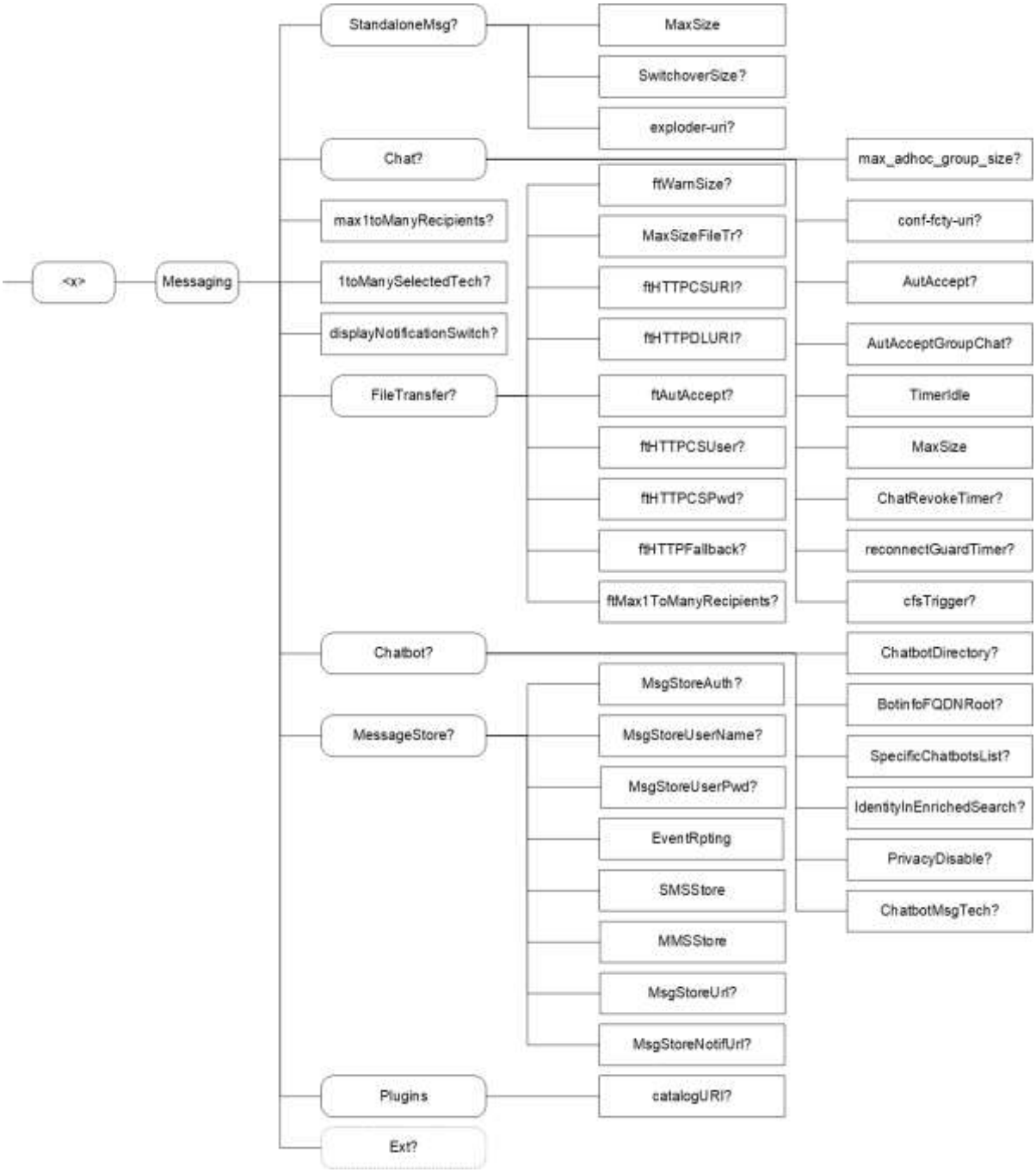



**Figure 20: RCS additions to the IM MO sub tree**

The associated HTTP configuration XML structure associated to the Messaging 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="MESSAGING">
        <characteristic type="StandaloneMsg">
                <parm name="MaxSize" value="X"/>
                <parm name="SwitchoverSize" value="X"/>
                <parm name="exploder-uri" value="X"/>
        </characteristic>
        <characteristic type="Chat">
                <parm name="max_adhoc_group_size" value="X"/>
                <parm name="conf-fcty-uri" value="X"/>
                <parm name="AutAccept" value="X"/>
                <parm name="AutAcceptGroupChat" value="X"/>
                <parm name="TimerIdle" value="X"/>
                <parm name="MaxSize" value="X"/>
                <parm name="ChatRevokeTimer" value="X"/>
                <parm name="reconnectGuardTimer" value="X"/>
                <parm name="cfsTrigger" value="X"/>
        </characteristic>
        <parm name="max1ToManyRecipients" value="X"/>
        <parm name="1toManySelectedTech" value="X"/>
        <parm name="displayNotificationSwitch" value="X"/>
        <characteristic type="FileTransfer">
                <parm name="ftWarnSize" value="X"/>
                <parm name="MaxSizeFileTr" value="X"/>
                <parm name="ftAutAccept" value="X"/>
                <parm name="ftHTTPCSURI" value="X"/>
                <parm name="ftHTTPDLURI" value="X"/>
                <parm name="ftHTTPCSUser" value="X"/>
                <parm name="ftHTTPCSPwd" value="X"/>
                <parm name="ftHTTPFallback" value="X"/>
                <parm name=" ftMax1ToManyRecipients" value="X"/>
        </characteristic>
        <characteristic type="Chatbot">
                <parm name="ChatbotDirectory" value="X"/>
                <parm name="BotinfoFQDNRoot" value="X"/>
                <parm name="SpecificChatbotsList" value="X"/>
                <parm name=”IdentityInEnrichedSearch” value=”X”/>
                <parm name=”PrivacyDisable” value=”X”/>
                <parm name=”ChatbotMsgTech” value=”X”/>
        </characteristic>
        <characteristic type="MessageStore">
                <parm name="MsgStoreUrl" value="X"/>
                <parm name="MsgStoreNotifUrl" value="X"/>
                <parm name="MsgStoreAuth" value="X"/>
                <parm name="MsgStoreUserName" value="X"/>
                <parm name="MsgStoreUserPwd" value="X"/>
                <parm name="EventRpting" value="X"/>
                <parm name="AuthArchive" value="X"/>
                <parm name="SMSStore" value="X"/>
                <parm name="MMSStore" value="X"/>
        </characteristic>
        <characteristic type="Plugins">
                <parm name="catalogURI" value="X"/>
        </characteristic>
        <characteristic type="Ext"/>
</characteristic>
```

**Table 122 : Messaging sub tree associated HTTP configuration XML structure**

Node: /<x>/Messaging

Under this interior node the RCS parameters related to the Messaging configuration are placed.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | node | Get |

**Table 123: IM MO sub tree addition IM node**

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-messaging:9.0*
- Associated HTTP XML characteristic type: “MESSAGING”

Node: /<x>/Messaging/StandaloneMsg

Interior node where parameters related to the RCS Text message and Multimedia message service based on CPM Standalone Messaging are provided

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 124: Messaging MO sub tree addition Standalone messaging node**

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-messaging:9.0:StandaloneMsg*
- Associated HTTP XML characteristic type: “StandaloneMsg”

Node: /<x>/Messaging/StandaloneMsg/MaxSize

Leaf node that represents the maximum authorised 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 125: Messaging MO sub tree addition parameters (MaxSize)**

- Values: <content maximum size in bytes>
- Post-reconfiguration actions: no additional actions at the time of re-configuration.
- Associated HTTP XML parameter ID: “MaxSize”

Node: /<x>/Messaging/StandaloneMsg/SwitchoverSize

Leaf node that determines the maximum message size of a Pager Mode CPM Standalone Message for the switch over from the Pager Mode to the Large Message Mode.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | int | Get |

**Table 126: Messaging MO sub tree addition parameters (SwitchoverSize)**

- Values: <switchover size in bytes>
- Post-reconfiguration actions: no additional actions at the time of re-configuration.

- Associated HTTP XML parameter ID: “SwitchoverSize”

Node: /<x>/Messaging/StandaloneMsg/exploder-uri

Leaf node that represents the address to be used as Request URI when sending a 1-to-Many CPM Standalone Message

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | chr | Get |

**Table 127: Messaging MO sub tree addition parameters (exploder-uri)**

- Values: the URI to be used as exploder URI
- Post-reconfiguration actions: no additional actions at the time of re-configuration.
- Associated HTTP XML parameter ID: “exploder-uri”

Node: /<x>/Messaging/Chat

Interior node where parameters related to the RCS Chat messaging service are provided

This node is not instantiated if the Service Provider does not enable Chat or Group Chat and is required to be instantiated otherwise.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | node | Get |

**Table 128: Messaging MO sub tree addition Chat messaging node**

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-messaging:9.0:Chat*
- Associated HTTP XML characteristic type: “Chat”

Node: /<x>/Messaging/Chat/max_adhoc_group_size

Leaf node that represent the maximum number of participants that are allowed to be invited for a new 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 |

**Table 129: Messaging MO sub tree addition parameters (max_adhoc_group_size)**

- Values: <the maximum number of invitees>
- Post-reconfiguration actions: no additional actions at the time of re-configuration.
- Associated HTTP XML parameter ID: “max_adhoc_group_size”

Node: /<x>/Messaging/Chat/conf-fcty-uri

Leaf node that represent the URI of the conference factory to be used as Request-URI when initiating a new Group Chat

It is required to be instantiated if a service provider enables Group Chat.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | chr | Get |

**Table 130: Messaging MO sub tree addition parameters (conf-fcty-uri)**

- Values: <the conference factory URI>
- Post-reconfiguration actions: no additional actions at the time of re-configuration.
- Associated HTTP XML parameter ID: “conf-fcty-uri”

Node: /<x>/Messaging/Chat/AutAccept

Leaf node that represent the automatic/manual chat session answer mode

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | bool | Get |

**Table 131: Messaging MO sub tree addition parameters (AutAccept)**

- Values: 0, 1
  0- Indicates manual answer mode
  1- Indicates automatic answer mode (default value)
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “AutAccept”

Node: /<x>/Messaging/Chat/AutAcceptGroupChat

Leaf node that represent the automatic/manual Group Chat session answer mode

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | bool | Get |

**Table 132: Messaging MO sub tree addition parameters (AutAcceptGroupChat)**

- Values: 0, 1
  0- Indicates manual answer mode
  1- Indicates automatic answer mode (default value)
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “AutAcceptGroupChat”

Node: /<x>/Messaging/Chat/TimerIdle

Leaf node that represents the timeout for a chat session in idle mode (when there is no chat user activity)

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | int | Get |

**Table 133: Messaging MO sub tree addition parameters (TimerIdle)**

- Values: <Timer value in seconds>
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “TimerIdle”

Node: /<x>/Messaging/Chat/MaxSize

Leaf node that represent the maximum size in bytes of a text Chat message that a user can enter in a 1-to-1 Chat or Group Chat session.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | int | Get |

**Table 134: Messaging MO sub tree addition parameters (MaxSize)**

- Values: < maximum size in bytes of a text Chat message that a user can enter>
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “MaxSize”

Node: /<x>/Messaging/Chat/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 135: Messaging 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: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “ChatRevokeTimer”

Node: /<x>/Messaging/Chat/reconnectGuardTimer

Leaf node that provides the minimum time the client shall be registered in IMS prior to sending a message revocation request for a chat message.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | int | Get |

**Table 136: Messaging MO sub tree addition parameters (reconnectGuardTimer)**

- Values: integer value defining the timeout to be used in seconds
  When set to 0, the client is not able to send MessageRevoke requests
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “reconnectGuardTimer”

Node: /<x>/Messaging/Chat/cfsTrigger

Leaf node that controls the client trigger to fallback to SMS when revocation procedures apply.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | bool | Get |

**Table 137: Messaging MO sub tree addition parameters (cfsTrigger)**

- Values:
  0 (default): the client shall fall back to SMS and send the Message Revoke request right after sending the SMS
  1: the client shall fall back to SMS right after receiving the MessageRevokeResponse request with the value of the result equal to “success”
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “cfsTrigger”

Node: /<x>/Messaging/max1toManyRecipients

Leaf node that provides the maximum number of contacts allowed to be included in the distribution list of the 1-to-Many messaging service.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | int | Get, Replace |

**Table 138: Messaging MO sub tree addition parameters (max1ToManyRecipients)**

- Values:
  0 (default): the 1-to-Many Messaging service is disabled
  1: the client can add unlimited number of recipients
  >1: integer value that indicates the maximum total number of recipients that can be included in the distribution list
- Post-reconfiguration actions:
  If the configuration parameter value transits from 0 to a value higher than zero, or the parameter is added with value higher than zero to the client configuration, the client shall unhide the UX elements for the management of the distribution lists.
  If the configuration parameter value transits from a value higher than 0 to 0 or the configuration parameter is removed, then the client shall hide the UX elements for the management of distribution lists. Any stored distribution lists shall be deleted.
  The new value of the configuration parameter shall be stored and applied from this time on.
- Associated HTTP XML parameter ID: “max1ToManyRecipients”

Node: /<x>/Messaging/1toManySelectedTech

Leaf node that allows selecting the 1-to-1 messaging technology to be used for the 1-to-Many messaging service.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | int | Get, Replace |

**Table 139: Messaging MO sub tree addition parameters (1toManySelectedTech)**

- Values:
  0 (default): SMS is selected
  1: RCS 1-to-1 Chat service is selected
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “1toManySelectedTech”

Node: /<x>/Messaging/displayNotificationSwitch

Leaf node that controls whether the sending of Display Notification is enabled/disabled on the recipient's client.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | int | Get, Replace |

**Table 140: Messaging MO sub tree addition parameters (displayNotificationSwitch)**

- Values: <Timer value in seconds>
  0 (default), Enable sending Display Notifications, The user may still disable it using the Display Notification setting
  1: Disable sending Display Notifications, The Display Notification setting is not available to the user.
- Post-reconfiguration actions: Start using the provided value the next time when receiving a message that requires Display Notification.
- Associated HTTP XML parameter ID: "displayNotificationSwitch"

Node: /<x>/Messaging/FileTransfer

Interior node where parameters related to the RCS File Transfer service are provided

This node is not instantiated if the Service Provider does not enable File Transfer and is required to be instantiated otherwise.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | node | Get |

**Table 141: Messaging MO sub tree addition File Transfer node**

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-messaging:9.0:FileTransfer*
- Associated HTTP XML characteristic type: "FileTransfer"

Node: /<x>/Messaging/FileTransfer/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.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | int | Get, Replace |

**Table 142: Messaging MO sub tree addition parameters (ftWarnSize)**

- Values: The file size threshold (in KB) or 0 to disable the warning
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: "ftWarnSize"

Node: /<x>/Messaging/FileTransfer/MaxSizeFileTr

Leaf node that represent the maximum authorised size of a file that can be sent using the RCS File Transfer service

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | int | Get |

**Table 143: Messaging MO sub tree addition parameters (MaxSizeFileTr)**

- Values: The maximum file size threshold (in KB) or 0 to disable the limit
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “MaxSizeFileTr”

Node: /<x>/Messaging/FileTransfer/ftAutAccept

Leaf node that describes whether a File Transfer invitation can be automatically accepted

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | bool | Get, Replace |

**Table 144: Messaging MO sub tree addition parameters (ftAutAccept)**

- Values:
  0, automatic acceptance is not possible (regardless of the size of the file).
  1 (default), 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: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “ftAutAccept”

Node: /<x>/Messaging/FileTransfer/ftHTTPCSURI

This parameter configures the URI of the HTTP Content Server where files are going to be uploaded on the originating side.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | chr | Get, Replace |

**Table 145: Messaging MO sub tree addition parameters (ftHTTPCSURI)**

- Values: URL of the Service Provider's File Transfer upload server
  Example:
  http://ftcontentserver.rcs. mnc001.mcc262.pup.3gppnetwork.org/path/?parm=foo
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “ftHTTPCSURI”

Node: /<x>/Messaging/FileTransfer/ftHTTPDLURI

This parameter provides the URL of Service Provider's File Transfer download server. If present, it enables the local file download for content URLs received via File Transfer or any other RCS messaging service.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | Chr | Get, Replace |

**Table 146: Messaging MO sub tree addition parameters (ftHTTPDLURI)**

- Values: URL of the Service Provider's File Transfer download server
  Example:
  https://dl.rcs.mnc001.mcc262.pup.3gppnetwork.org/path?parm=foo
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “ftHTTPDLURI”

Node: /<x>/Messaging/FileTransfer/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).

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | chr | No Get, No Copy |

**Table 147: Messaging MO sub tree addition parameters (ftHTTPCSUser)**

- Values: The string containing **user value**.
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “ftHTTPCSUser”

Node: /<x>/Messaging/FileTransfer/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).

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | chr | No Get, No Copy |

**Table 148: Messaging MO sub tree addition parameters (ftHTTPCSPwd)**

- Values: The string containing **password value**.
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “ftHTTPCSPwd”

Node: /<x>/Messaging/FileTransfer/ftHTTPFallback

Leaf node that describes the operator's default setting client switch to control the user dialog for File Transfer fallback to SMS.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | bool | Get, Replace |

**Table 149: Messaging MO sub tree addition parameters (ftHTTPFallback)**

- Values: 0, 1
  - 0, MMS is used (default)
  - 1, Text message with a link is used
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “ftHTTPFallback”

Node: /<x>/Messaging/FileTransfer/ftMax1ToManyRecipients

Leaf node that provides the maximum number of recipients allowed for a File Transfer to multiple recipients.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | int | Get, Replace |

**Table 150: Messaging MO sub tree addition parameters (ftMax1ToManyRecipients)**

- Values:
  0 (default): the client can send a File Transfer to an unlimited number of recipients
  Positive integer value: indicates the maximum total number of recipients a File Transfer can be sent to.
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “ftMax1ToManyRecipients”

Node: /<x>/Messaging/Chatbot

Interior node where there are filled parameters related to RCS Chatbot Functionality

This node is not instantiated if the Service Provider does not enable Chat.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | node | Get |

**Table 151: Messaging MO sub tree addition Chatbot node**

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-messaging:9.0:Chatbot*
- Associated HTTP XML characteristic type: “Chatbot”

Node: /<x>/Messaging/Chatbot/ChatbotDirectory

Leaf node that represents the URL address from which the client should retrieve the list of Chatbots.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | chr | Get |

**Table 152: Messaging MO sub tree addition parameters (ChatbotDirectory)**

- Values: the URL for retrieving the list for Chatbots provided by the Service Provider
- Post-reconfiguration actions: no additional actions at the time of re-configuration.
- Associated HTTP XML parameter ID: “ChatbotDirectory”

Node: /<x>/Messaging/Chatbot/BotinfoFQDNRoot

Leaf node that represents the root part of the FQDN to be used by the client to compose the botinfo URL as defined in section 3.6.4.1.1.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | chr | Get |

**Table 153: Messaging MO sub tree addition parameters (BotinfoFQDNRoot)**

- Values: string indicating the root part of the FQDN to be used by the client to compose the botinfo URL
- Post-reconfiguration actions: no additional actions at the time of re-configuration.
- Associated HTTP XML parameter ID: “BotinfoFQDNRoot”

Node: /<x>/Messaging/Chatbot/SpecificChatbotsList

Leaf node that represents the URL address from which the client should retrieve the lists with URIs of Chatbots requiring specific management.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | chr | Get |

**Table 154: Messaging MO sub tree addition parameters (SpecificChatbotsList)**

- Values: the URL for retrieving the list of Chatbots requiring specific management
- Post-reconfiguration actions: no additional actions at the time of re-configuration.
- Associated HTTP XML parameter ID: “SpecificChatbotsList”

Node: /<x>/Messaging/Chatbot/IdentityInEnrichedSearch

Leaf node that determines whether the i query parameter defined in section 3.6.3.1 is included in the client to Service Provider Chatbot Directory requests when the user setting to enrich the search is enabled.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | int | Get |

**Table 155: Messaging MO sub tree addition parameters (IdentityInEnrichedSearch)**

- Values: 0,1

  0- the i query parameter is not included (default value)
  1- the i query parameter is included

- Post-reconfiguration actions: no additional actions at the time of re-configuration.
- Associated HTTP XML parameter ID: “IdentityInEnrichedSearch”

Node: /<x>/Messaging/Chatbot/PrivacyDisable

Leaf node that represents the authorisation for the user to request anonymization for a Chatbot session.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | bool | Get |

**Table 156: Messaging MO sub tree addition parameters (PrivacyDisable)**

- Values: 0,1

  0- Indicates that Anonymization is enabled (default value)
  1- Indicates that Anonymization is disabled

- Post-reconfiguration actions: no additional actions at the time of re-configuration.
- Associated HTTP XML parameter ID: “PrivacyDisable”

Node: /<x>/Messaging/Chatbot/ChatbotMsgTech

Leaf node that controls the messaging technology for Chatbot conversation from the client.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | int | Get |

**Table 157: Messaging MO sub tree addition parameters (ChatbotMsgTech)**

- Values: 0,1,2,3

  0: Chatbot Services are disabled
  1 (default): Chatbot Services are enabled only with 1-to-1 Chatbot Chat Sessions
  2: Chatbot Services are enabled with both 1-to-1 Chatbot Chat Sessions and 1-to-1 Chatbot Standalone Messaging
  3: Chatbot Services are enabled only with 1-to-1 Chatbot Standalone Messaging

- Post-reconfiguration actions:

  - If the value transits from "0" or "3" to "1" or "2", then the client shall register or re-register in IMS to add the feature tags for Chatbot Communication with Sessions and Chat defined in section 2.4.4, if conditions allow.
  - If the value transits from "1" or "2" to "0" or "3" and if the client is registered in IMS for Chatbot Communication with Sessions as defined in section 2.4.4, then the client shall initiate an IMS de-registration or re-registration to remove the feature tags for Chatbot Communication with Sessions defined in section 2.4.4. If the GROUP CHAT AUTH client configuration parameter is not set to 1, the client shall also remove the feature tag for Chat defined in section 2.4.4.
  - If the value transits from "0" or "1" to "2" or "3", then the client shall register or re-register in IMS to add the feature tags for Standalone Messaging and for Chatbot Communication with Standalone Messaging defined in section 2.4.4, if conditions allow.
  - If the value transits from "2" or "3" to "0" or "1" and if the client is registered in IMS for Chatbot Communication with Standalone Messaging as defined in section 2.4.4, then the client shall initiate an IMS de-registration or re-registration to remove the feature tags for Chatbot Communication with Standalone Messaging defined in section 2.4.4. If the STANDALONE MSG AUTH client configuration parameter is not set to 1, the client shall also remove the feature tags for Standalone Messaging defined in section 2.4.4.

- Associated HTTP XML parameter ID: “ChatbotMsgTech”

Node: /<x>/Messaging/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 158: Messaging MO sub tree addition Message Store node**

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-messaging:9.0:MessageStore*
- Associated HTTP XML characteristic type: “MessageStore”

Node: /<x>/Messaging/MessageStore/MsgStoreUrl

Leaf node that represents the URL address of the Message Store Server

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | chr | Get |

**Table 159: Messaging MO sub tree addition parameters (MsgStoreUrl)**

- Values: the URL for accessing the Message Store Server
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “MsgStoreUrl”

Node: /<x>/Messaging/MessageStore/MsgStoreNotifUrl

Leaf node that represents the URL address of the Message Store notification server

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | chr | Get |

**Table 160: Messaging MO sub tree addition parameters (MsgStoreNotifUrl)**

- Values: the URL for accessing the Message Store notification server
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “MsgStoreNotifUrl”

Node: /<x>/Messaging/MessageStore/MsgStoreAuth

Leaf node indicating whether a Message Store server is available and whether HTTP Basic or other authentication mechanism requested by the Message Store server shall be used.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | int | Get |

**Table 161: Messaging MO sub tree addition parameters (MsgStoreAuth)**

- Values: 0, 1, 2
  0- Indicates that the Message Store server is not enabled (default)
  1- Indicates that Message Store server is enabled and the client shall initiate HTTP Basic authentication using user name and password.

2- Indicates that the Message Store server is enabled and the client shall perform the authentication mechanism requested by the Message Store server.

- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “MsgStoreAuth”

Node: /<x>/Messaging/MessageStore/MsgStoreUserName

Optional leaf node that represents the User Identity information used by the Message Store Client to access the subscriber Message Store account on the Message Store Server

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | chr | No Get, No Copy |

**Table 162: Messaging MO sub tree addition parameters (MsgStoreUserName)**

- Values: <username assigned to the user for access to the Message Store Server>
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “MsgStoreUserName”

Node: /<x>/Messaging/MessageStore/MsgStoreUserPwd

Optional 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 Message Store account on the Message Store Server

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | chr | No Get, No Copy |

**Table 163: Messaging MO sub tree addition parameters (MsgStoreUserPwd)**

- Values: <password assigned to the user for access to the Message Store Server>
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “MsgStoreUserPwd”

Node: /<x>/Messaging/MessageStore/EventRpting

Optional leaf node that can be used to inform the Message Store Client whether to directly set flags in the Message Store or whether to indicate to the Messaging Server that it should set flags in the Message Store on behalf of the client. If not instantiated, 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 164: Messaging MO sub tree addition parameters (EventRpting)**

- Values: 0, 1
  0- Indicates that the client shall set flags in the Message Store as needed via an Message Store connection (default)
  1- Indicates that the client shall make use of the Event Reporting framework as described in section 4.1.16.4 and 4.1.16.5 when no Message Store connection exists

so that the Messaging Server may set the flags in the Message Store on behalf of the client
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “EventRpting”

Node: /<x>/Messaging/MessageStore/AuthArchive

Optional leaf node that can be used to enable the Message Store Client to archive messages.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | bool | Get |

**Table 165: Messaging MO sub tree addition parameters (AuthArchive)**

- Values: 0, 1
  0- Indicates that the archive service is disabled (default)
  1- Indicates that archive service is enabled and thus the client may archive messages.
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “AuthArchive”

Node: /<x>/Messaging/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 166: Messaging 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 Default folder
  2- The device shall store every sent and received SMS and shall not attempt to correlate with the Common Message Store in the RCS Default folder.
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “SMSStore”

Node: /<x>/Messaging/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 167: Messaging 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: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “MMSStore”

Node: /<x>/Messaging/Plugins

Interior node where parameters related to the Plug-ins are provided

This node is not instantiated if the Service Provider does not provide Plug-ins and is required to be instantiated otherwise.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | node | Get |

**Table 168: Messaging MO sub tree addition Plugins node**

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-messaging:9.0:Plugins*
- Associated HTTP XML characteristic type: “Plugins”

Node: /<x>/Messaging/Plugins/catalogURI

This parameter configures the URI used to construct the Catalog retrieval URL used to access the Plug-info server for retrieving or refreshing the Catalog.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | chr | Get, Replace |

**Table 169: Messaging MO sub tree addition parameters (catalogURI)**

- Values: The string containing the URI used to construct the Catalog retrieval URL used to access the Plug-info server for initially retrieving or refreshing the Catalog.
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- Associated HTTP XML parameter ID: “catalogURI”

Node: /<x>/Messaging/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 170: IM MO sub tree addition Service Provider Extension Node**

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-messaging:9.0:Ext*
- Associated HTTP XML characteristic type: “Ext”

### A.2.5. 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:

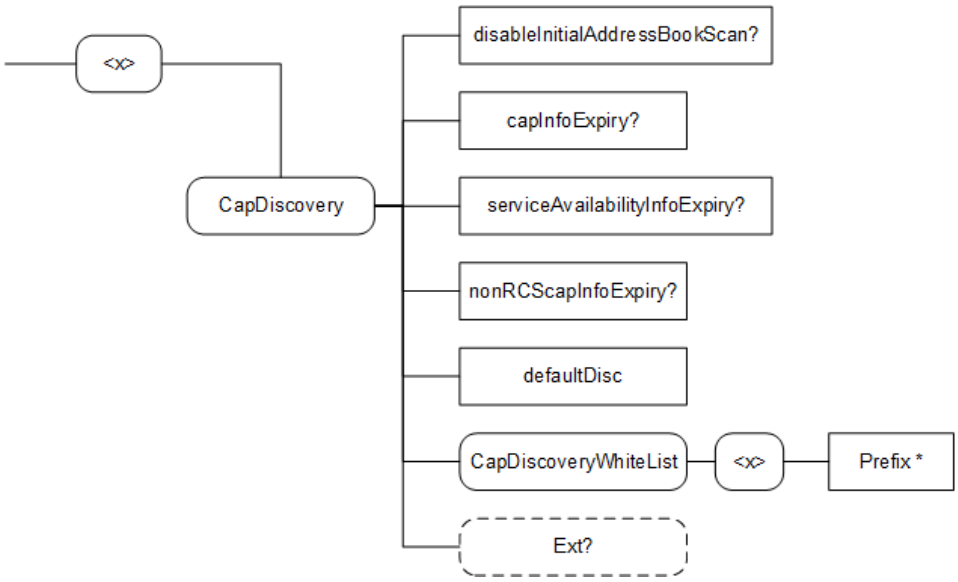



**Figure 21: RCS additions, capability sub tree**

The associated HTTP configuration XML structure is presented in the table below:

```
<characteristic type="CAPDISCOVERY">
        <parm name="disableInitialAddressBookScan" value="X"/>
        <parm name="capInfoExpiry" value="X"/>
        <parm name="nonRCScapInfoExpiry" value="X"/>
        <parm name="serviceAvailabilityInfoExpiry" 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 171 : 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 172: Capability MO sub tree addition capability discovery node**

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-icapdis:9.0*
- Associated HTTP XML characteristic type: “CAPDISCOVERY”

Node: /<x>/CapDiscovery/disableInitialAddressBookScan

Leaf node that describes whether the device/client should when it is first started, perform a capability exchange for all contacts in the address book

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | int | Get, Replace |

**Table 173: Capability MO sub tree addition parameters (disableInitialAddressBookScan)**

- Values:
  0, The scan of the address book shall be done
  1, The scan of the address book shall not be done.
- Post-reconfiguration actions: if changed from 1 to 0 and no prior address book scan was done, the client shall initiate an address book scan.
- Associated HTTP XML parameter ID: “disableInitialAddressBookScan”

Node: /<x>/CapDiscovery/capInfoExpiry

Leaf node that describes the validity of the capability information cached in the terminal

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | int | Get, Replace |

**Table 174: Capability MO sub tree addition parameters (capInfoExpiry)**

- Values: The validity time in seconds, 0 indicates that there is no expiry.
- Post-reconfiguration actions:
  - When changing from a positive value to 0, the already cached capability information shall be considered to never expire
  - When changing from 0 to a positive value, the client shall consider the expiry of the currently cached capabilities to be the newly configured value and start monitoring from the moment of reconfiguration
  - When increasing the value, the newly configured expiry value shall be applied to newly cached capabilities only

  - When decreasing the value, the client shall consider the expiry of the currently cached capabilities to be the newly configured value and start monitoring from the moment of reconfiguration

- Associated HTTP XML parameter ID: “capInfoExpiry”

Node: /<x>/CapDiscovery/serviceAvailabilityInfoExpiry

Leaf node that describes the validity of the availability information cached in the terminal in seconds.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | int | Get, Replace |

**Table 175: Capability MO sub tree addition parameters (serviceAvailabilityInfoExpiry)**

- Values: The time in seconds
- Post-reconfiguration actions:
  - When increasing the value, the newly configured expiry value shall be applied to newly cached availability information only
  - When decreasing the value, the client shall consider the expiry of the currently cached capabilities to be the new value and start monitoring from the moment of reconfiguration
- Associated HTTP XML parameter ID: “serviceAvailabilityInfoExpiry”

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 176: Capability MO sub tree addition parameters (nonRCScapInfoExpiry)**

- Values: The time in seconds.
- Post-reconfiguration actions: No additional actions at the time of reconfiguration.
- 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 | ZeroOrOne | int | Get, Replace |

**Table 177: 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
2, the mechanism employed for capability discovery will be disabled.

- Post-reconfiguration actions: When the value is changed from 0 or 2 to 1, the client shall also publish its capabilities as defined in section 2.6.1.2.2.
- Associated HTTP XML parameter ID: “defaultDisc”

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 178: Capability MO sub tree addition Capability Discovery White List node**

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-icapdis:9.0: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 179: Capability MO sub tree addition CapDiscoveryAllowedPrefixes node**

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-icapdis:9.0: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 180: Capability MO sub tree addition parameters (Prefix)**

- Values:
  The value can contain either a single prefix or a single regular expression for matching with phone numbers. If the phone number matches the prefix or the regular expression, then the phone number shall be considered for capability and new user discovery. If the phone number does not match the prefix or regular expression, then the client shall match the phone number with the value of the next configuration parameter in the capability discovery white list. If the phone number matches with none of the values of the "Prefix" configuration parameters in the capability discovery

white list, then the phone number shall not be considered for capability and new user discovery.

To match a phone number with prefixes and regular expressions, the client shall remove visual separators and white space from the input phone number string.

- The configuration parameter contains a prefix if the value consists of a number string, optionally preceded by a "+" character. The client shall match the phone number and the prefix contained in the configuration parameter by string match. The phone number matches, if there is a full match of the prefix with the beginning of the phone number string.
  Examples:
  +446
  +4479
  00446
  004479
  06
  079
- The configuration parameter contains a regular expression if the value starts with the "!" character. The subsequent string shall be interpreted by the client using Portable Operating System Interface (POSIX) extended regular expression (see [POSIX]). The phone number matches, if the application of the regular expression results in a non-empty string.
  Examples:
  !^(0044|0)(6|79)
  !\+44(6|79)\d*

- Post-reconfiguration actions: If a Prefix is no longer included in the list, the client shall remove capability and service availability information that is stored for contacts whose phone number matches the removed prefix from the cache.
- 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 181: Capability MO sub tree addition Service Provider Extension Node**

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-icapdis:9.0:Ext*
- Associated HTTP XML characteristic type: “Ext”

### A.2.6. 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:

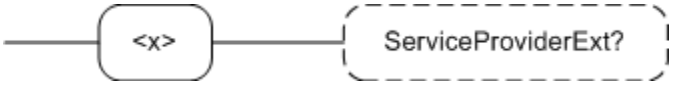



**Figure 22: RCS additions, Service Provider Extensions sub tree**

The associated HTTP configuration XML structure is presented in the table below:

```
<characteristic type="SERVICEPROVIDEREXT"/>
```

**Table 182 : 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 183: Service Provider Extensions MO sub tree addition node**

- Values: N/A
- Type property of the node is: *urn:gsma:mo:rcs-sp:9.0*
- Associated HTTP XML characteristic type: “SERVICEPROVIDEREXT”

## A.3. Other Management Objects

### A.3.1. Overview

The configuration XML document for RCS services includes configuration parameters from other than the Management Object defined in this document. The following table provides an overview of other Management Objects applicable for RCS services.

| Management Object | Application Characteristic Reference | AppID Value | RCS Usage |
|---|---|---|---|
| OMA Management Object for Presence SIMPLE 2.0 [PRESENCE2MO] | Embedded in RCS Application characteristic as defined in section A.2.3 | n/a | Section A.1.2.1 |
| 3GPP IMS Management Object [3GPP TS 24.167] | [PRD-RCC.15] | urn:oma:mo:ext-3gpp-ims:1.0 | Section A.1.6.1 |

| Management Object | Application Characteristic Reference | AppID Value | RCS Usage |
|---|---|---|---|
| Non-Access Stratum (NAS) configuration Management Object [3GPP TS 24.368] | NAS configuration MO DDF as defined in [3GPP TS 24.368] via transformation as defined in Annex A of [PRD-RCC.14] | urn:oma:mo:ext-3gpp-nas-config:1.0 | Section A.1.14.2 |

**Table 184: Other Management Objects applicable for RCS**

For an overview of other Management Objects applicable for VoLTE, SMS over IP, VoWiFi, ViLTE and ViWiFi refer to [PRD-IR.92], [PRD-IR.51] and [PRD-IR.94].

### A.3.2. IMS sub tree additions

The IMS Core network configuration for RCS shall be provided in a configuration XML document by use of the IMS MO defined in [3GPP TS 24.167] via the provisioning document defined in [PRD-RCC.15].

Additional parameters applicable for RCS have been defined for the IMS MO in [PRD-RCC.15].

This section extends the IMS MO defined in [3GPP TS 24.167] with additional RCS specific parameters that are added to the existing IMS MO extension tree defined in [PRD-RCC.15].

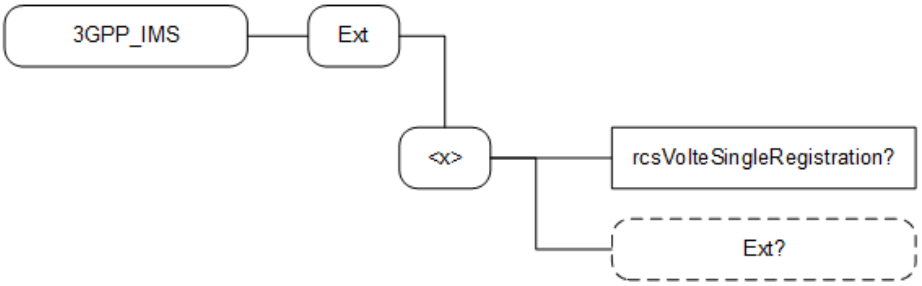



**Figure 23: Additions to the IMS MO sub tree**

Following parameters have been defined:

Node: <x>/rcsVolteSingleRegistration

Where <x> corresponds to the <x> node below the Ext node of the IMS sub tree defined in [PRD-RCC.15].

Leaf node that describes the behaviour regarding the instantiation of the IMS stack in devices supporting all RCS services (i.e. including Multimedia Telephony and SMSoIP). It is also used to control the APN selection on devices that are not enabled for VoLTE or VoWiFi as described in section 2.8.1.4.

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | int | Get, Replace |

**Table 185: IMS Configuration sub tree addition parameters (rcsVolteSingleRegistration)**

- Values:

  - 0, the device shall follow a dual registration approach (transition solution) where RCS services other than Mutlimedia Telephony and SMSoIP use a separate registration from the VoLTE/VoWiFi one.
  - 1 (default if not provided), the device shall follow a single registration (target solution) for all RCS services (i.e. including Multimedia Telephony and SMSoIP) services.
  - 2, the device shall follow a single registration for all RCS services (i.e. including Multimedia Telephony and SMSoIP) when in the home network or EPC integrated WiFi, and shall follow a dual registration when roaming (transition solution).

- Post-reconfiguration actions: If the value of the configuration parameter changes, then the device shall de-register RCS services from IMS, establish bearers if required for the new value of the configuration parameter and shall register or re-register in IMS to update registration for RCS services in accordance with the new value of the configuration parameter and release bearers from the old configuration if no longer needed.
- Associated HTTP XML characteristic type: “rcsVolteSingleRegistration”

The non-normative provisioning document structure for the IMS MO extension is presented in the table below

```
<characteristic type="3GPP_IMS">
        …
        <characteristic type="Ext">
              <charcteristic type="GSMA">
                    …
                    <parm name="rcsVolteSingleRegistration" value="X"/>
                    ...
                    <characteristic type="Ext"/>
              </characteristic>
        </characteristic>
</characteristic>
```

**Table 186: Provisioning document structure of the IMS MO extension**

## A.4. Configuration XML document structure and examples

### A.4.1. HTTP configuration XML structure

The provisioning documents applicable for RCS are embedded in a configuration XML document as depicted in the non-normative document structure shown in Table 187.

```
<?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">                   <!-- This section is OPTIONAL -->
                <parm name="token" value="X"/>
        </characteristic>
        <!-- Potentially optional characteristics e.g. MSG, User and Access Control, see [PRD-RCC.14]-->

        <characteristic type="APPLICATION">
                <parm name=”AppID” value=”urn:oma:mo:ext-3gpp-ims:1.0”/>
                <characteristic type="3GPP_IMS">
                        <parm name="AppID" value="ap2001"/>
                        <parm name="Name" value="RCS IMS Settings"/>
                        …       <!-- see [PRD-RCC.15] -->
                         <characteristic type="Ext">
                                <characteristic type="GSMA">
                                        <parm name=”AppRef” value=”IMS-Setting”/>
                                        <parm name="rcsVolteSingleRegistration" value="X"/>
                                        …       <!-- see section [PRD-RCC.15] -->
                                </characteristic>
                        </characteristic>
                </characteristic>
        </characteristic>

        <characteristic type="APPLICATION">
                <parm name=”AppID” value=” urn:oma:mo:ext-3gpp-nas-config:1.0”/>
                <characteristic type="NODE">
                        … <!-- See [3GPP TS 24.368] -->
                </characteristic>
        </characteristic>

        <characteristic type="APPLICATION">
                <parm name="AppID" value="ap2002"/>
                <parm name="To-AppRef" value="IMS-Setting"/>
                <characteristic type="SERVICES">
                        …       -- See section A.2.2
                </characteristic>
                <characteristic type="PRESENCE">
                        …       -- See section A.2.3
                </characteristic>
                <characteristic type="MESSAGING">
                        …       -- See section A.2.4
                </characteristic>
                <characteristic type="CAPDISCOVERY">
                        …       -- See section A.2.5
                </characteristic>
                <characteristic type="SERVICEPROVIDEREXT">
                        …       -- See section A.2.6
```

```
            </characteristic>
        </characteristic>
</wap-provisioningdoc>
```

**Table 187: Complete configuration XML structure**

### A.4.2. Configuration XML document example

This section provides a non-normative configuration XML document example.

```
<?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">                   <!-- This section is OPTIONAL -->
                <parm name="token" value="X"/>
        </characteristic>
        <!-- Potentially additional, optional characteristics such as MSG, User and Access Control -->
        <!-- see [PRD-RCC.14] -->
        <characteristic type="APPLICATION">
                <parm name=”AppID” value=”urn:oma:mo:ext-3gpp-ims:1.0”/>
                <characteristic type="3GPP_IMS">
                        <parm name="AppID" value="ap2001"/>
                        <parm name="Name" value="RCS IMS Settings"/>
                        …                       <!-- see [PRD-RCC.15] -->
                        <characteristic type="Ext">
                                <characteristic type="GSMA">
                                        <parm name="AppRef" value="RCS-IMS-Settings"/>
                                        …                       <!-- see [PRD-RCC.15] -->
                                </characteristic>
                        </characteristic>
                </characteristic>
        </characteristic>
        <characteristic type="APPLICATION">
                <parm name=”AppID” value=” urn:oma:mo:ext-3gpp-nas-config:1.0”/>
                <characteristic type="NODE">
                        <characteristic type="3GPP_PS_data_off">
                                <characteristic type="Exempted_service_list">
                                        <parm name=”Device_management_over_PS” value=”X”/>
                                        <parm name="Bearer_independent_protocol" value="X"/>
                                </characteristic>
                                <characteristic type="Exempted_service_list_roaming">
                                        <parm name=”Device_management_over_PS” value=”X”/>
                                        <parm name="Bearer_independent_protocol" value="X"/>
                                </characteristic>
                                <!-- See [3GPP TS 24.368] -->
                        </characteristic>
                </characteristic>
        </characteristic>
        <characteristic type="APPLICATION">
                <parm name="AppID" value="ap2002"/>
                <parm name="To-AppRef" value="RCS-IMS-Settings"/>
                <characteristic type="SERVICES">
                        <parm name=”SupportedRCSVersions” value=”X”/>
                        <parm name=”SupportedRCSProfileVersions” value=”X”/>
                        <parm name="ChatAuth" value="X"/>
                        <parm name="GroupChatAuth" value="X"/>
                        <parm name="ftAuth" value="X"/>
                        <parm name="standaloneMsgAuth" value="X"/>
-- Continues in the next table --
```

**Table 188: Complete RCS configuration XML example (1/3)**

```
-- Follows from previous table –

                <parm name="geolocPushAuth" value="X"/>
                <parm name="rcsIPVoiceCallAuth" value="X"/>
                <parm name="rcsIPVideoCallAuth" value="X"/>
                <parm name="composerAuth" value="X"/>
                <parm name="sharedMapAuth" value="X"/>
                <parm name="sharedSketchAuth" value="X"/>
                <parm name="postCallAuth" value="X"/>
                <characteristic type="Ext">
                        <characteristic type="DataOff">
                            <parm name="rcsMessagingDataOff" value="X"/>
                            <parm name="fileTransferDataOff" value="X"/>
                            <parm name="mmsDataOff" value="X"/>
                            <parm name="contentShareDataOff" value="X"/>
                            <parm name="preAndPostCallDataOff" value="X"/>
                            <parm name="provisioningDataOff" value="X"/>
                            <parm name="syncDataOff" value="X"/>
                            <characteristic type="Ext"/>
                        </characteristic>
                    </characteristic>
            </characteristic>
             <characteristic type="PRESENCE">
                <parm name="client-obj-datalimit" value="X"/>
                <parm name="source-throttlepublish" value="X"/>
                <parm name="RLS-URI" value="X"/>
            </characteristic>
            <characteristic type="MESSAGING">
                <characteristic type="StandaloneMsg">
                        <parm name="MaxSize" value="X"/>
                        <parm name="SwitchoverSize" value="X"/>
                        <parm name="exploder-uri" value="X"/>
                </characteristic>
                <characteristic type="Chat">
                        <parm name="max_adhoc_group_size" value="X"/>
                        <parm name="conf-fcty-uri" value="X"/>
                        <parm name="AutAccept" value="X"/>
                        <parm name="AutAcceptGroupChat" value="X"/>
                        <parm name="TimerIdle" value="X"/>
                        <parm name="MaxSize" value="X"/>
                        <parm name="ChatRevokeTimer" value="X"/>
                        <parm name="reconnectGuardTimer" value="X"/>
                        <parm name="cfsTrigger" value="X"/>
                </characteristic>
                <parm name="max1ToManyRecipients" value="X"/>
                <parm name="1toManySelectedTech" value="X"/>
                <parm name="displayNotificationSwitch" value="X"/>
                <characteristic type="FileTransfer">
                        <parm name="ftWarnSize" value="X"/>
                        <parm name="MaxSizeFileTr" value="X"/>
                        <parm name="ftAutAccept" value="X"/>
                        <parm name="ftHTTPCSURI" value="X"/>
                        <parm name="ftHTTPDLURI" value="X"/>
                        <parm name="ftHTTPCSUser" value="X"/>
                        <parm name="ftHTTPCSPwd" value="X"/>
-- Continues in the next table --
```

**Table 189: Complete RCS Configuration XML example (2/3)**

```
-- Follows from previous table –

                        <parm name="ftHTTPFallback" value="X"/>
                        <parm name="ftMax1ToManyRecipients" value="X"/>
                </characteristic>
                <characteristic type="Chatbot">
                        <parm name="ChatbotDirectory" value="X"/>
                        <parm name="BotinfoFQDNRoot" value="X"/>
                        <parm name="SpecificChatbotsList" value="X"/>
                        <parm name=”IdentityInEnrichedSearch” value=”X”/>
                        <parm name=”PrivacyDisable” value=”X”/>
                        <parm name=”ChatbotMsgTech” value=”X”/>
                </characteristic>
                <characteristic type="MessageStore">
                        <parm name="MsgStoreUrl" value="X"/>
                        <parm name="MsgStoreNotifUrl" value="X"/>
                        <parm name="MsgStoreAuth" value="X"/>
                        <parm name="MsgStoreUserName" value="X"/>
                        <parm name="MsgStoreUserPwd" value="X"/>
                        <parm name="EventRpting" value="X"/>
                        <parm name="AuthArchive" value="X"/>
                        <parm name="SMSStore" value="X"/>
                        <parm name="MMSStore" value="X"/>
                </characteristic>
                <characteristic type="Plugins">
                        <parm name="catalogURI" value="X"/>
                </characteristic>
                <characteristic type="Ext"/>
        </characteristic>
        <characteristic type="CAPDISCOVERY">
                <parm name="disableInitialAddressBookScan" value="X"/>
                <parm name="capInfoExpiry" value="X"/>
                <parm name="nonRCScapInfoExpiry" value="X"/>
                <parm name="serviceAvailabilityInfoExpiry" value="X"/>
                <parm name="defaultDisc" value="X"/>
                <characteristic type="CapDiscoveryWhitelist">
                        <characteristic type="CapDiscoveryAllowedPrefixes">
                                <parm name="Prefix1" value="X"/>
                                <parm name="Prefix2" value="X"/>
                        </characteristic>
                </characteristic>
                <characteristic type="Ext"/>
        </characteristic>
        <characteristic type="SERVICEPROVIDEREXT"/>
    </characteristic>
</wap-provisioningdoc>
```

**Table 190: Complete RCS configuration XML example (3/3)**

# Annex B: Additional diagrams

## B.1. Chat and store and forward diagrams

### B.1.1. Store and forward: Receiver offline

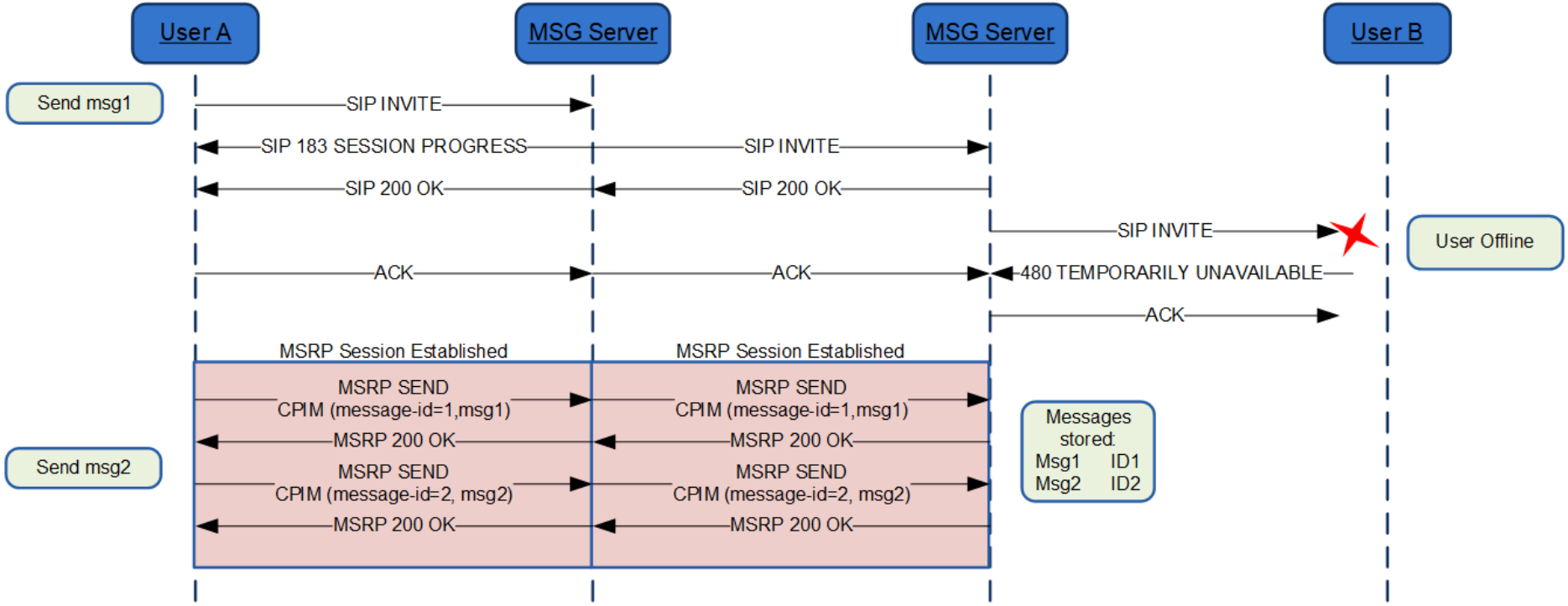



**Figure 24: Store and forward: Receiver offline***

*: Check NOTES 1, 5 and 8 in section B.1.11.

### B.1.2. Store and forward: Message deferred delivery with sender still on an active Chat session

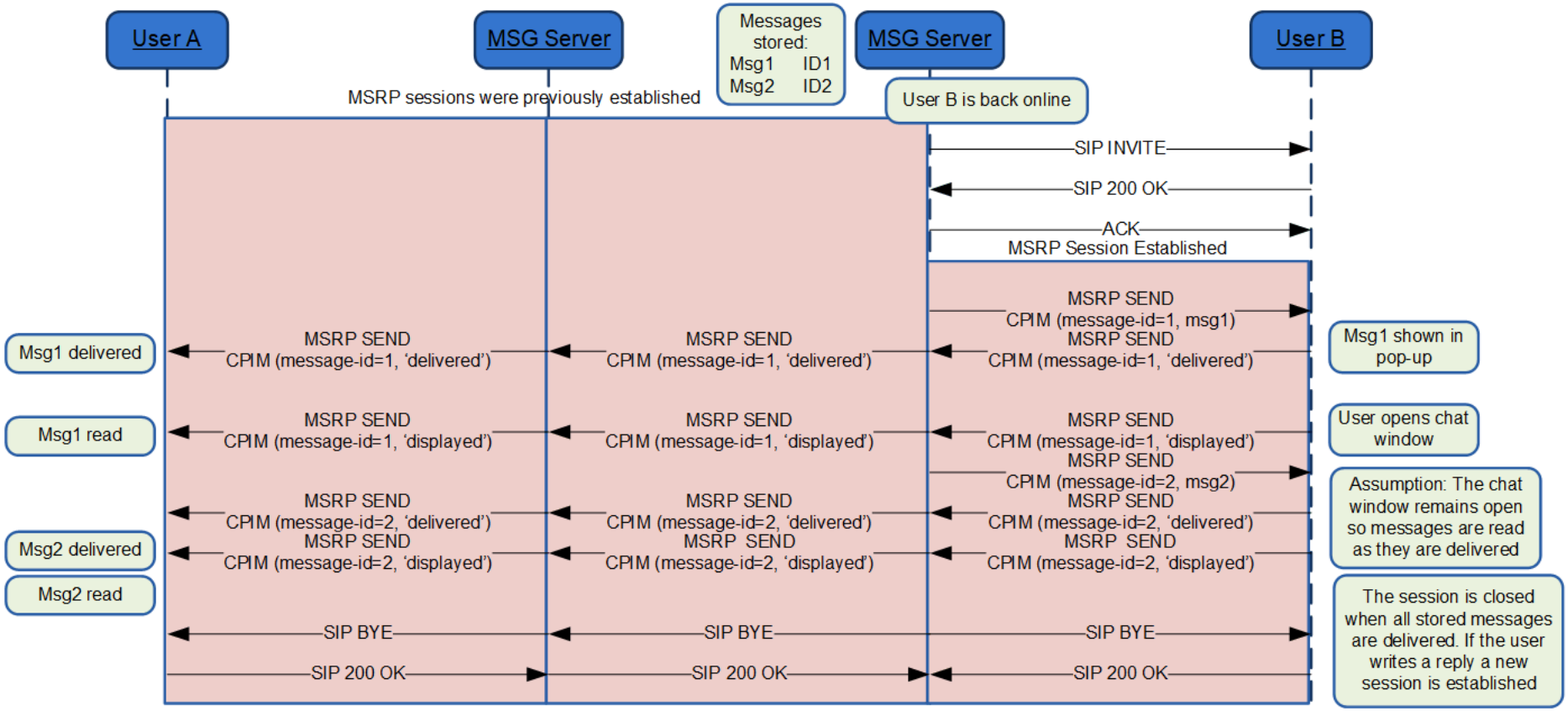



**Figure 25: Store and forward: Message(s) deferred delivery with a sender still on an MSRP session***

*: Check NOTES 1, 2, 3, 6, 7 and 8 in section B.1.11

### B.1.3. Store and forward: Message deferred delivery with sender online

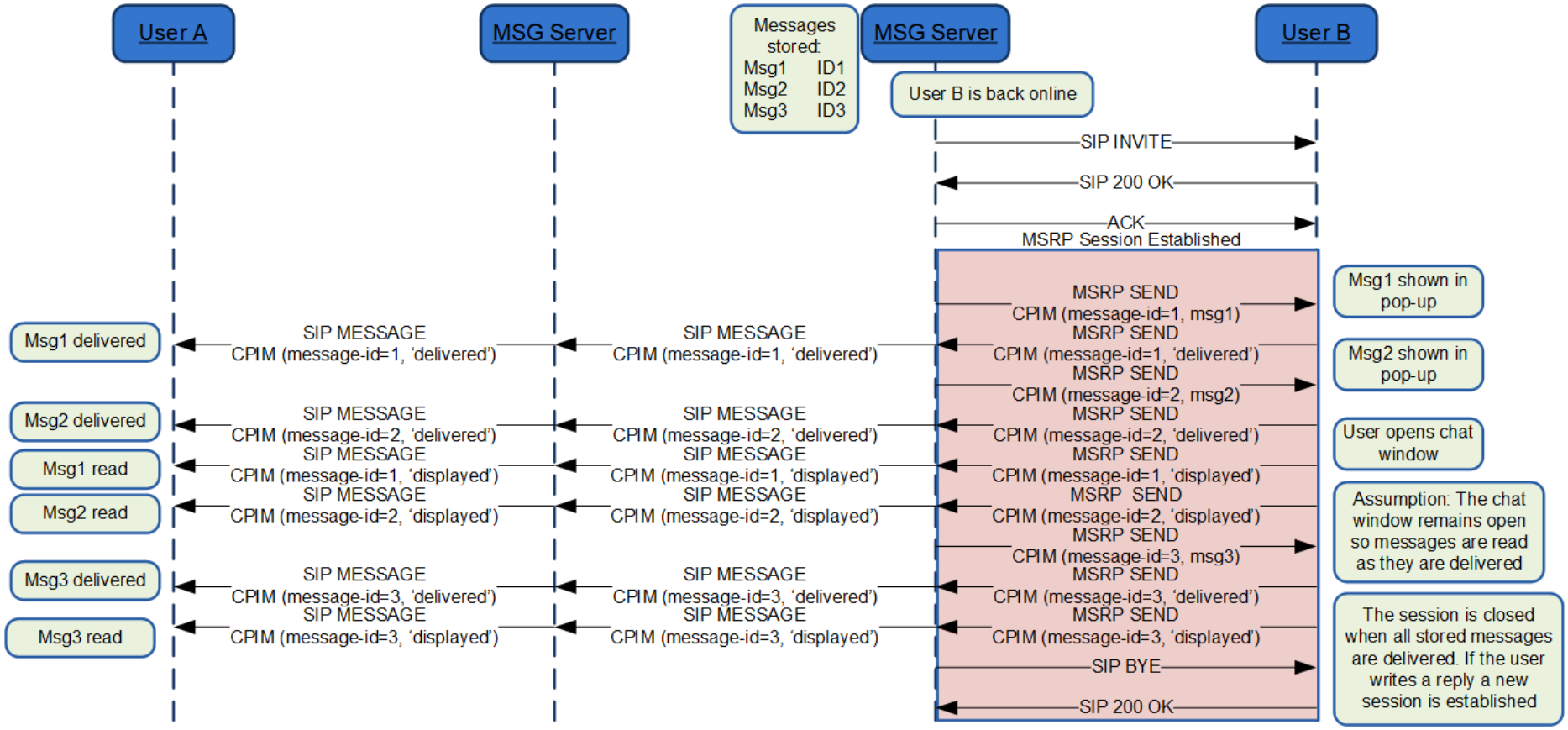



**Figure 26: Store and forward: Message deferred delivery with sender online ***

*: Check NOTES 1, 2, 3, 6, 7 and 8 in section B.1.11

### B.1.4. Store and forward: Message deferred delivery with sender offline (delivery notifications)

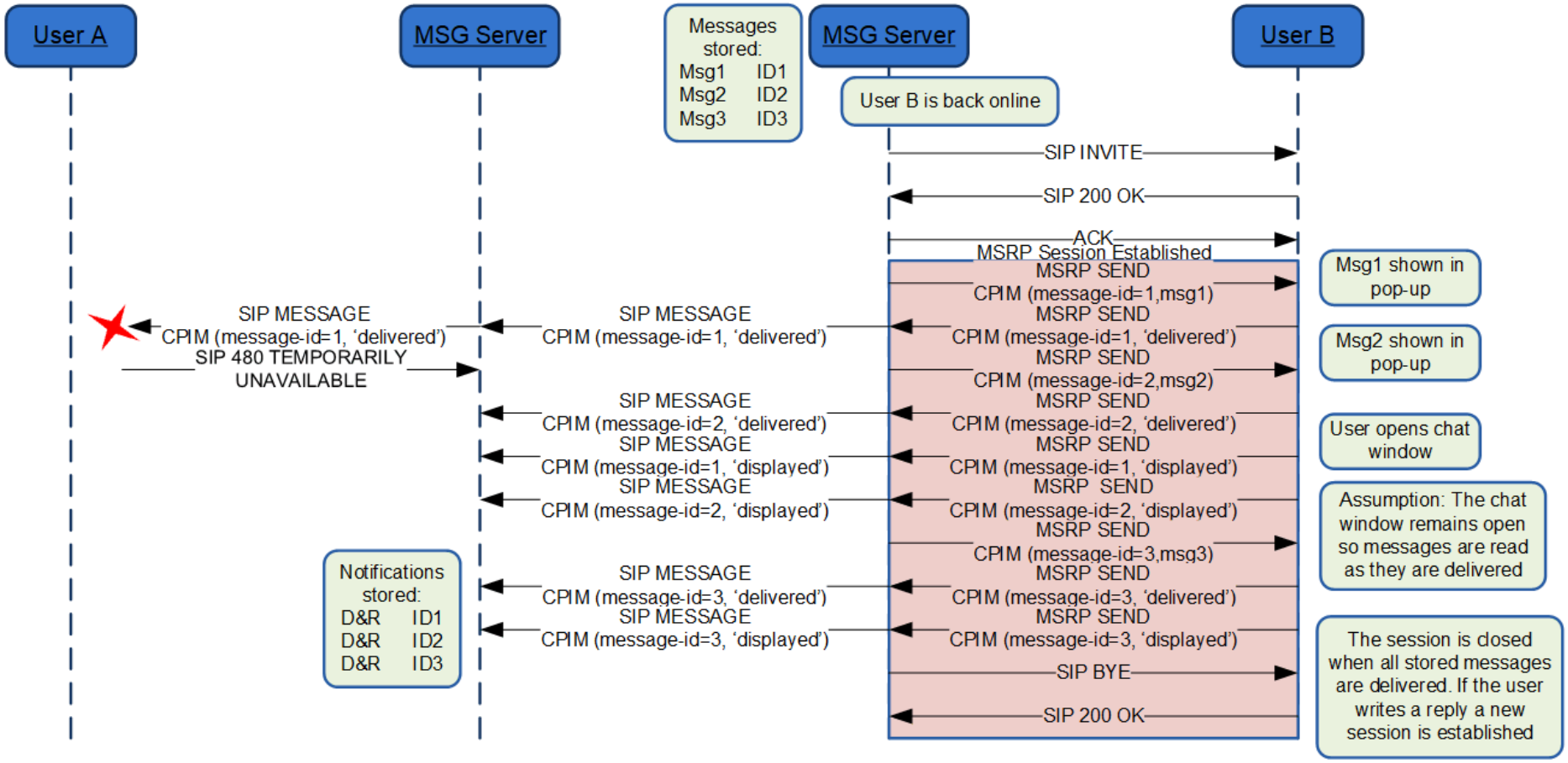



**Figure 27: Store and forward: Message(s) deferred delivery with a sender offline (delivery notifications)***

*: Check NOTE 1, 4, 6, 7 and 8 in section B.1.11

### B.1.5. Store and forward: Notifications deferred delivery

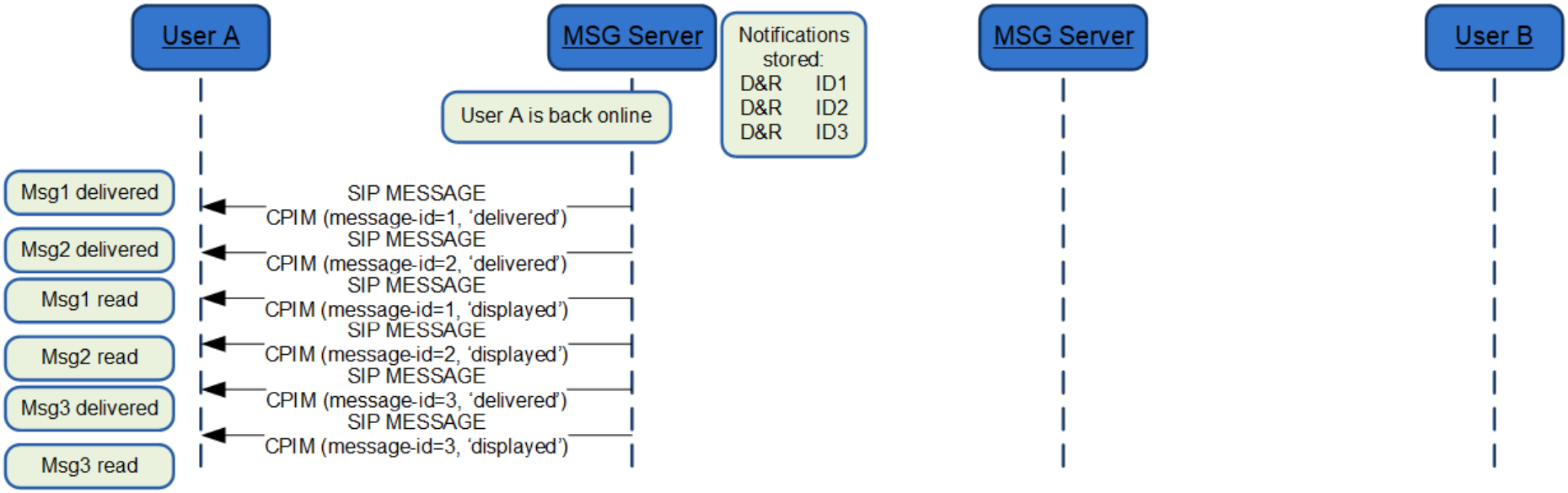



**Figure 28: Store and forward: Notification(s) deferred delivery***

*: Check NOTES 1 and 8 in section B.1.11.

### B.1.6. Network Interworking to SMS/MMS

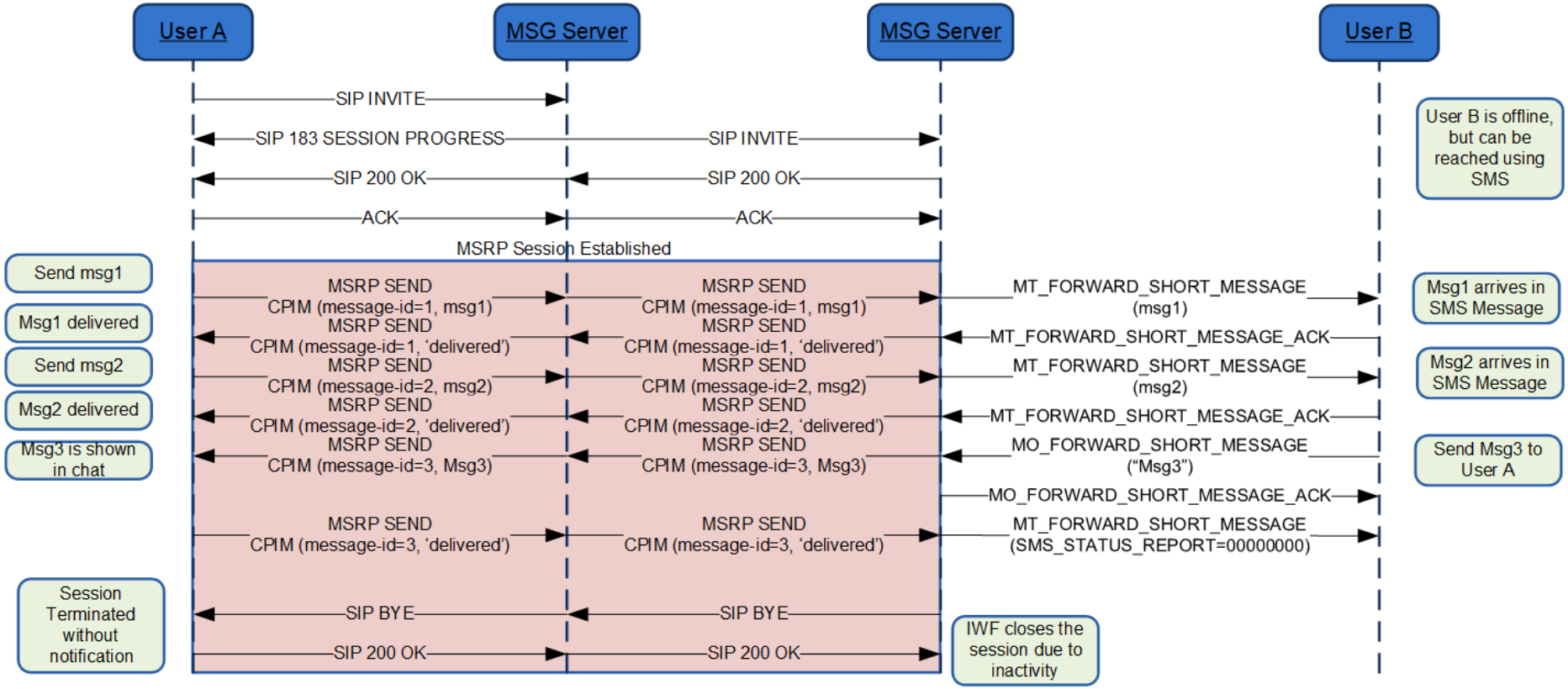



**Figure 29: Interworking: Automatic acceptance on behalf of the SMS/MMS user***

*: Check NOTES 1, 8 and 9 in section B.1.11.

### B.1.7. Message Revoke: Successful Request

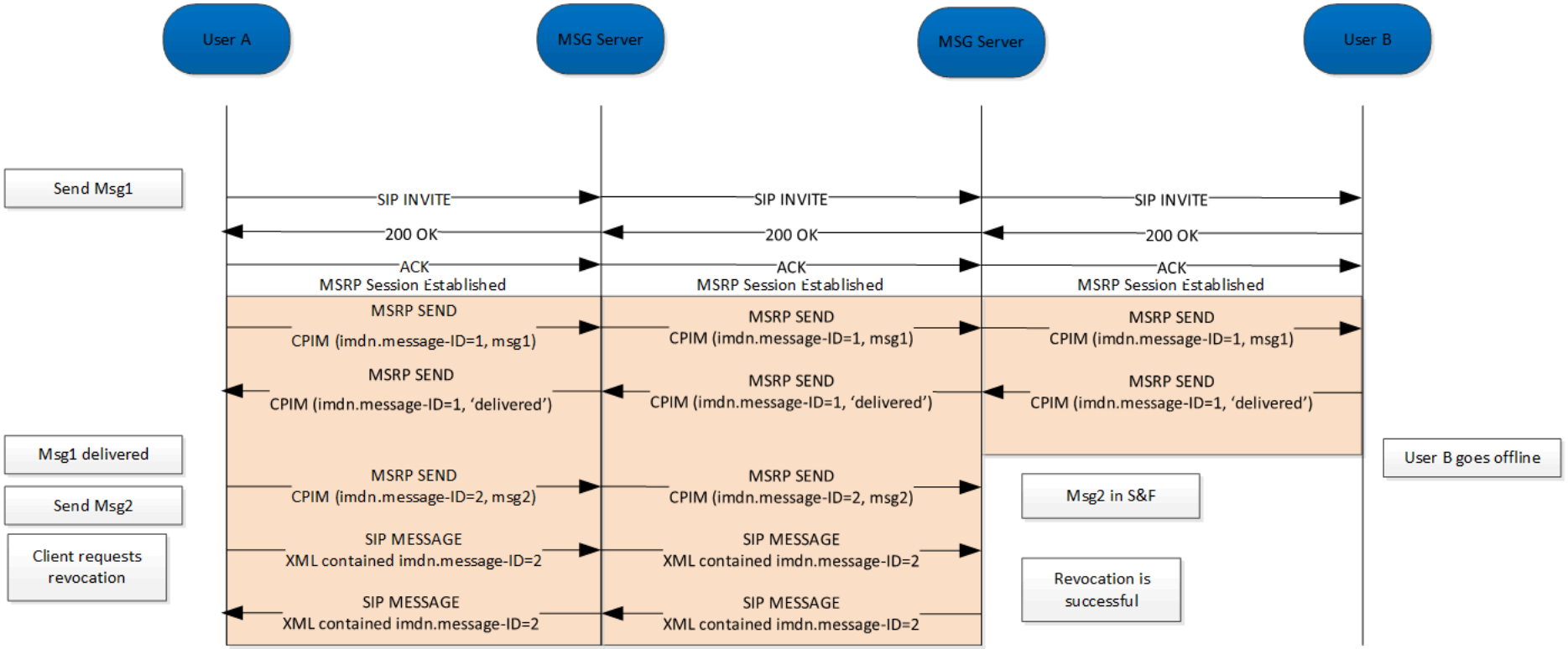



**Figure 30: Message Revoke, Successful request***

*: Check NOTES 1 and 8 in section B.1.11

### B.1.8. Message Revoke: Failed Request

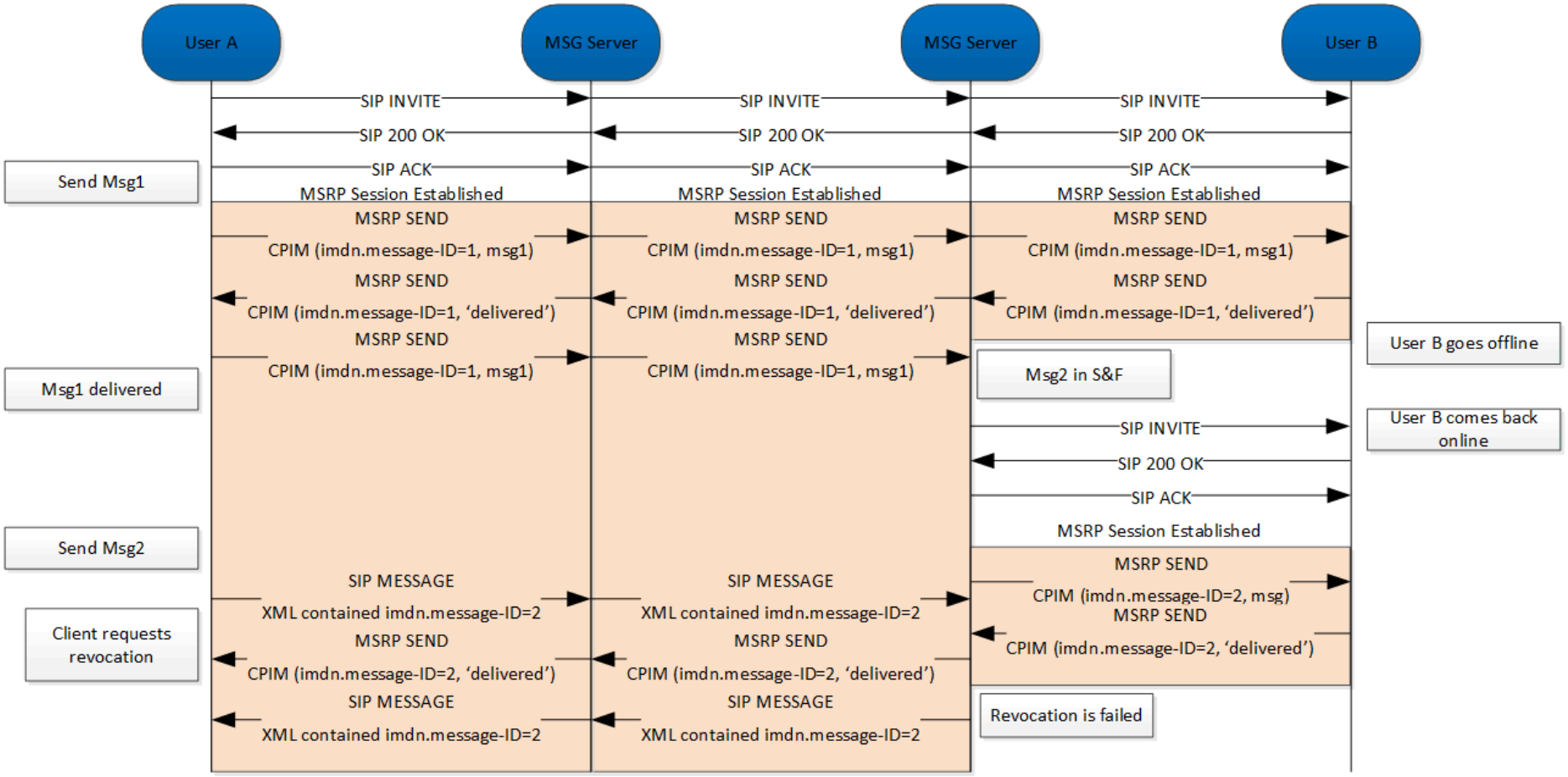



**Figure 31: Message Revoke, Failed request***

*: Check NOTES 1 and 8 in section B.1.11

### B.1.9. Deliver Stored Group Chat Messages while Chat is idle

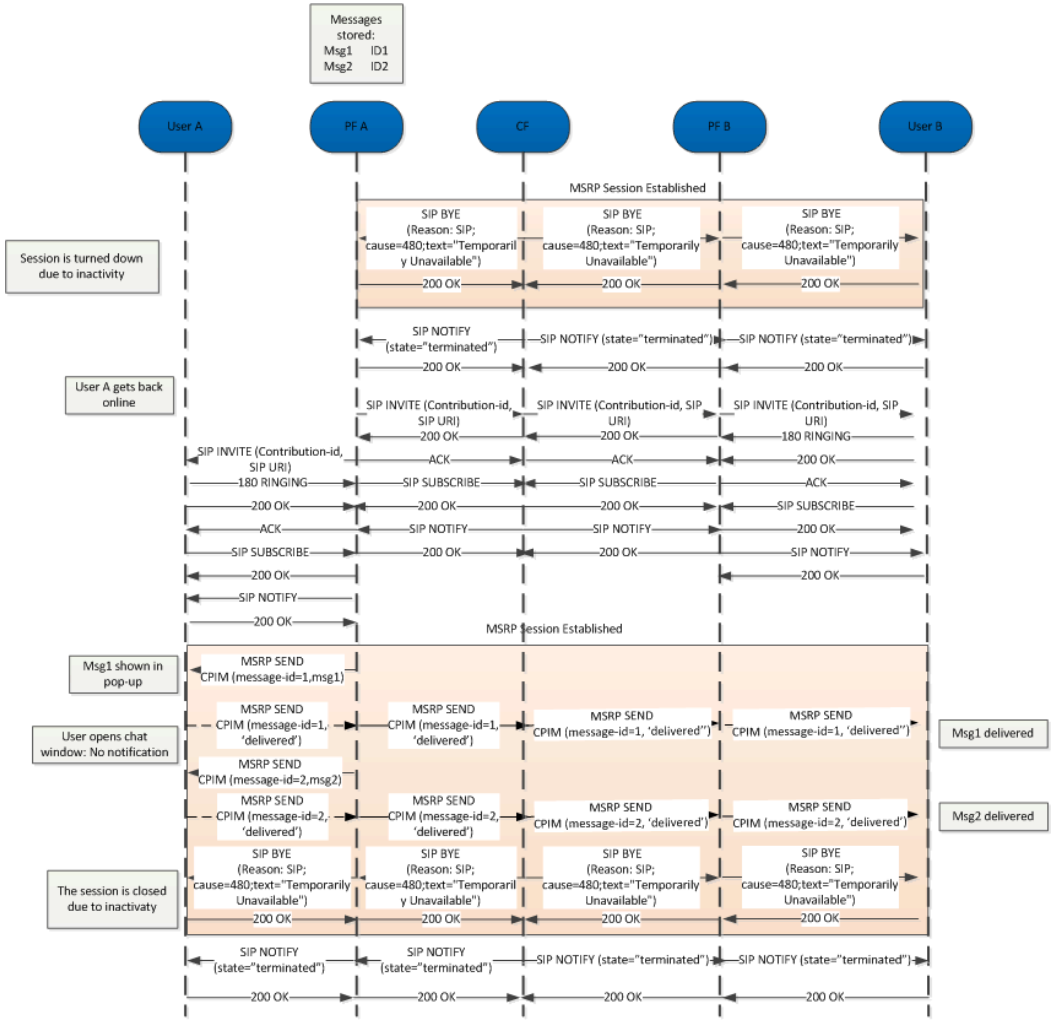



**Figure 32: Deliver Group Chat Messages while Chat is idle***

*: Check NOTES 1, 8, 10, 11 and 12 in section B.1.11.

### B.1.10. Multi-device

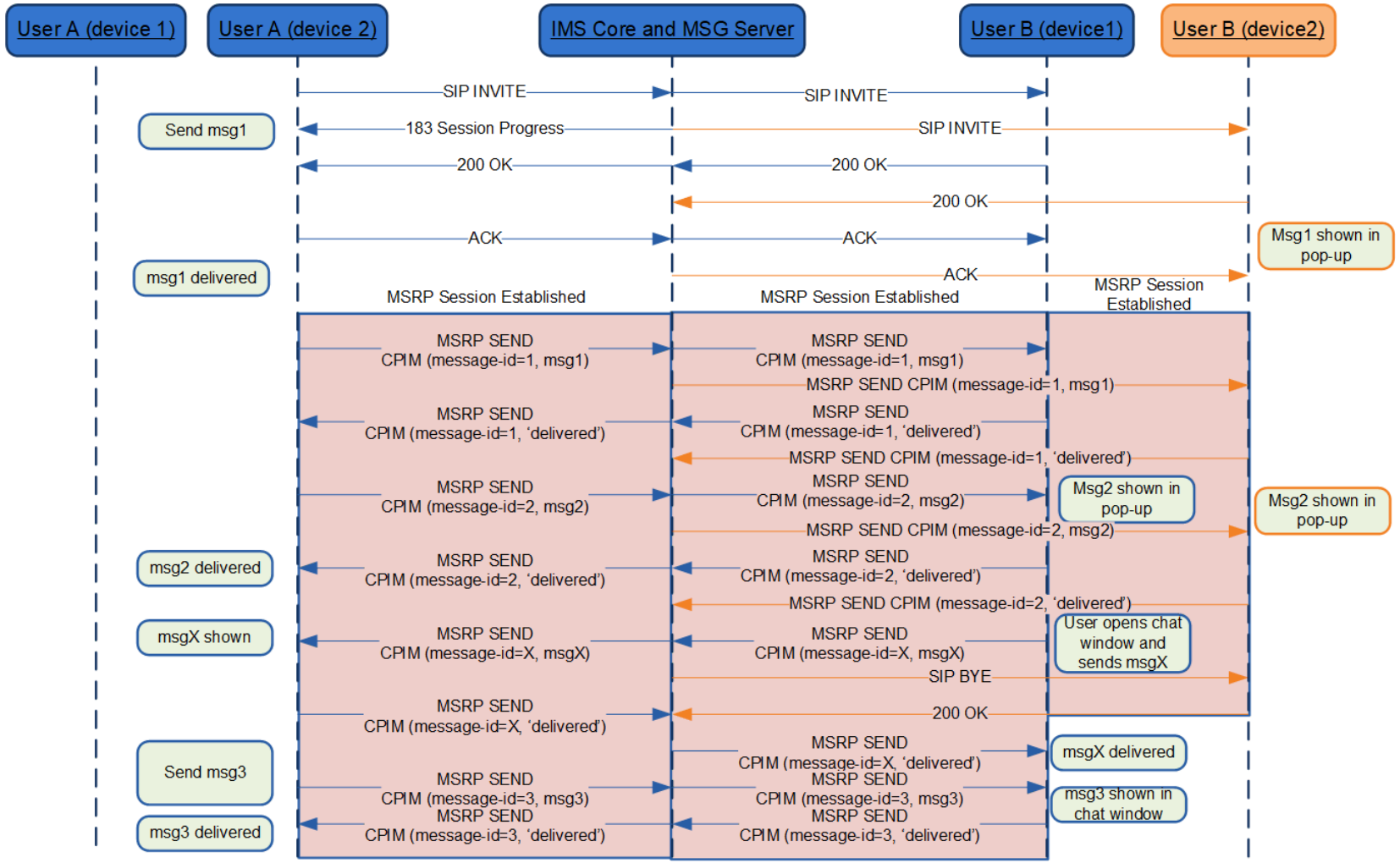



**Figure 33: Rejoin in active Chat which is torn down due to inactivity***

*: Check NOTES 1, 8, 13, 14 and 15 in section B.1.11.

### B.1.11. 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.1.8, B.1.9 and B.1.10): 200 OK responses to SIP MESSAGE and MSRP SEND messages are omitted for clarity.
- NOTE 2 (B.1.2 and B.1.3): In a multidevice scenario, a delivery notification received from User B might not end up on User A's device that sent the message. However this is not an issue, since all User A's devices will eventually receive the delivery notification upon synchronising with the Common Message Store.
- NOTE 3 (B.1.2 and B.1.3): 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 recognise 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 4 (B.1.4): The flow assumes that the Messaging Server does not attempt to deliver subsequent IMDNs if delivery of a first one failed shortly before..
- NOTE 5 (B.1.1): 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 6 (B.1.2, B.1.3 and B.1.4): 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 7 (B.1.2, B.1.3 and B.1.4): 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 8 (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 and B.1.10): 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 9 (B.1.6): The flow shows interworking with SMS, but the flow in the SIP/MSRP part of the figure also applies when interworking with MMS.
- NOTE 10 (B.1.9): As per sections 3.2.4.10 and 3.2.4.15.
- NOTE 11 (B.1.9): 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 12 (B.1.9): The flow assumes that no display notifications were requested.
- NOTE 13 (B.1.10): The diagram 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 14 (B.1.10): To support this case forking in the terminating side needs to be done at the Messaging Server using the mechanisms defined in section 2.10.2 as forking in the IMS core will lead to a race condition.
- NOTE 15 (B.1.10): The behaviour shown to close the session to the other devices when there is activity from one receiving device is just one of the possible behaviours. Alternatively, the session to those devices could be maintained and all sent and received messages, notifications and events could be delivered to all devices with appropriate indications as described in [RCS-CPM-CONVFUNC-ENDORS].

## B.2. Restful Message Store Flows (informative)

### B.2.1. Client Initialization and Synchronization using RESTful Approach

There are two types of initialization that the client must consider:

1. First Time Sync - Running the application for the first time having either had no previous contact with MESSAGE STORE SERVER or having had to clear and reinitialize the local message store.
2. Steady State Sync – Ongoing sync with the Message Store Server. This happens during normal operation; it also covers the case when connections time out, network signal is lost, the application or phone is restarted etc.

The overall initialization process for a client may look as follows.

1. Client ensures there is a valid OAuth Token to use with MESSAGE STORE SERVER,
2. starts the Notification process and is listening for updates to MESSAGE STORE SERVER, initiates the First Time Sync process if appropriate
3. Waits for notifications to arrive, i.e. initiates the Steady State Sync process.
4. The Notification process on the client polls for notifications that have arrived from MESSAGE STORE SERVER. Note again that client ensures there is a valid OAuth token for use by both it and the Notification thread. This means that the Notification thread must wait if it gets an authentication error until the Worker thread has obtained a new OAuth token

The following sections discuss in more detail how the First Time and Steady State Sync functions work.

#### *B.2.1.1. First Time Sync Process*

First Time Sync performs the following to get the local store and MESSAGE STORE SERVER in sync with one another. This is in two stages and result in both stores having the same set of messages.

- Download existing messages from MESSAGE STORE SERVER to the handset.
- Upload any handset messages that are not present in MESSAGE STORE SERVER.

Once First Time Sync has completed then Steady State Sync can start.

Two key notes to consider:

1. Since First Time Sync can take a significant amount of time to complete, the process may be interrupted. The handset may lose network access or be stopped.
2. Any user interactions with messages on the handset such as reading or deleting messages need to be queued up until the message in question has been correlated with MESSAGE STORE SERVER.

Since messages are returned newest first, the handset may choose to synchronize only the most recent messages, saving local storage space.

**Recommended Approach:**

To ensure a graceful re-sync restart, the handset must store the following information persistently:

- The Index of the latest notification returned by a poll – to check for missed notifications.
- The latest RestartToken value – used to restart the polling process if interrupted.
- The latest /objects/operations/search cursor - used to restart the search process if interrupted.
- The time at which the existing subscription (if any) times out - used to know when the lifetime of the subscription needs updating.
- A list of the messages on the handset that have not been correlated with MESSAGE STORE SERVER.
- Whether First Time Sync message upload has completed or not - to decide whether Steady State Sync should run or not.

The client application should perform the following steps:

- Perform the user interaction needed to begin first time sync – gaining acceptance of terms, account provisioning, etc. After this point, all synchronization is now performed within the new threads.
- Start listening for Notifications and wait until it has an active long poll call to MESSAGE STORE SERVER.
- Submit a search request to the MESSAGE STORE SERVER with a blank cursor, or the stored cursor if restarting an interrupted sync.

  *Begin processing the queue of the response to the search request:*

- Use the request arrival time and number of downloaded messages to determine the pause before issuing another request. The overall rate for this sync should be one message/second to limit handset, network and MESSAGE STORE SERVER load.
- For a search request: download the messages using the objects/operations/search endpoint using the specified cursor.

  - The first search uses a blank cursor, and subsequent searches use the cursor supplied by the last search.
  - The number of messages to be downloaded in each search is specified using the maxEntries field – the recommended value is 10.

  On receiving a response to a search:

  - Iterate through the message headers and correlate with locally stored messages. If a correlationId is available for the message, use it to correlate message. Otherwise, correlate using the correlationTag.

    - If a correlating message is found, then update the local message, and remove from the list of uncorrelated messages.

- Be aware that in rare cases, when relaying on a correlationTag, multiple matches may occur. In this case, follow the guidance provided in Dealing with Hash Collisions.
    - If not found then store the message locally as an uncorrelated message.
- For small messages, such as SMS messages, all the information on the message will be retrieved in the search. For larger messages, the client will need to retrieve the content by downloading the payload parts of the message – but the client will not need to download the message headers as these are returned by the search.
- Add a new request to the worker queue. If the cursor is not blank, add a search request with the new cursor. If it is blank, messages have finished downloading; add an upload request for the first uncorrelated message.

- For a notification request, update the local message store with the changed objects. Event notifications contain new, changed or deleted objects. How the client processes such a notification depends on whether the message already exists on the client:
    - For messages that already exist on the local store, use the lastModSeq attribute to determine whether to update the local copy of the message.
    - Save the latest lastModSeq of every object and folder. If the network copy has a lastModSeq greater than the local one, the client needs to update the local copy to match.
    - This does not need any further requests; the updated information is contained in the poll response.
    - When updating the local copy of an uncorrelated message in response to a sync, remove that message from the list of uncorrelated messages.
    - For messages that do not have a local version, the client will need to download the message from the message store.

- For a request from the application thread to modify a message
    - If the message has been correlated with MESSAGE STORE SERVER, the thread should make the update to MESSAGE STORE SERVER.
    - If the message has not yet been correlated, the update must be delayed until the message has been correlated (which may only be after the First Time Sync process is complete).

#### *B.2.1.2. Notification Flow*

- The Client shall start a notification channel and subscription.
- Begin the Steady State long poll flow.
    - Issue poll request. This must be on a separate network connection (pipelining requests on the same HTTP connection is not recommended in long polling.)
    - If the poll returns with an empty notification list or no content, it has timed out - reissue the poll request.

- Otherwise, the poll returns with one or more event notifications. In this case
    - Check the index of the notifications returned against the saved index. The smallest of the new indices should be one greater than the saved index; if this is not the case, the client has missed a notification.
    - Otherwise, reissue the poll.
    - Add the notifications returned to the worker queue for processing.
- The client is responsible for ensuring that the subscription does not time out; the client must keep track of the remaining subscription duration and periodically extend it before it expires.
- The MESSAGE STORE SERVER service refreshes notification channels automatically when the client performs a long poll. If the client experiences an outage, the notification channel and/or subscription it is using may expire; for details on how to handle this case, see section 7.3.1 of [CPM-MSGSTOR-REST].

#### *B.2.1.3. Steady State Synchronization*

The purpose of the Steady State Sync process is to update the handset with changes made to the subscriber's mailbox.

In overview, the client waits for notifications of incoming messages using a polling request, and may concurrently process already received notifications, uploading and downloading messages and performing all correlation of MESSAGE STORE SERVER with the local handset store.

Recommended Approach:

- For a notification request, the client shall update the local message store with the changed objects. Event notifications contain new, changed or deleted objects. How the client processes such a notification depends on whether the message already exists on the client:
    - For a ResetBox notification see [CPM-MSGSTOR-REST] section 5.1.4.2.1; in summary the MESSAGE STORE SERVER client application data (local message cache, counters, etc.) should be emptied and the client should perform a first time sync with MESSAGE STORE SERVER.
    - Perform correlation steps provided for objects with accompanied by a correlationId and correlation Tag. If a correlationId does not exist, in which case a correlationTag is provided. If both are provided, the client shall use the correlationId.
    - For messages that already exist on the local store (in other words, for message that are successfully correlated), client shall use the lastModSeq attribute to determine whether it needs to update its local copy of the message.
        - The client should save the latest lastModSeq of every object and folder. If the network copy has a lastModSeq greater than the local one, the client needs to update the local copy to match.

- This does not need any further requests; the updated information is contained in the poll response.

- Outside of First Time Sync, a client should never download message content from the message store. The message may reside in the client's uncorrelated sync objects list, but the client should never present it to the user.

- For a request from the application thread to modify a message, the thread should also make the update to MESSAGE STORE SERVER.

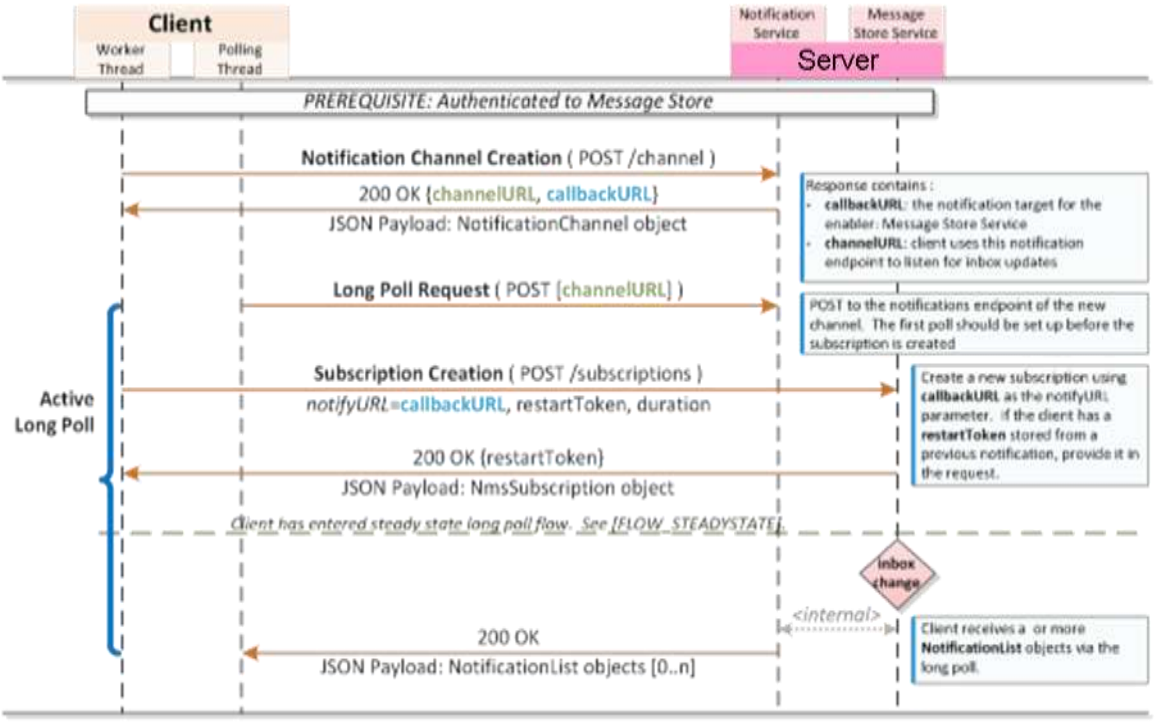
### B.2.2. RESTful Notification Events

For Notification about storage changes, the events in Table 191 are combined into an NmsEventList before being sent to clients with an appropriate subscription.

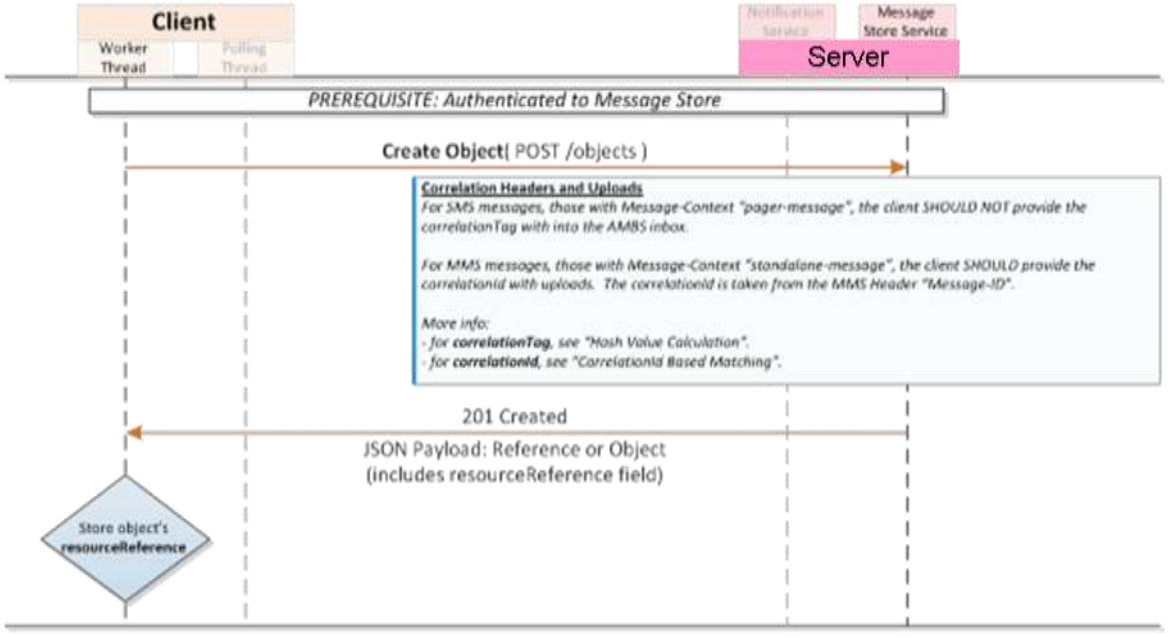
| Element | Type | Optional | Description |
|---|---|---|---|
| deletedObject | DeletedObject | Choice | Reference to the user-deleted object |
| deletedFolder | DeletedFolder | Choice | Reference to the user-deleted folder |
| expiredObject | DeletedObject | Choice | Reference to the expired object |
| expiredFolder | DeletedFolder | Choice | Reference to the expired folder. |
| changedObject | ChangedObject | Choice | Reference to the new or changed object. |
| changedFolder | ChangedFolder | Choice | Reference to the new or changed folder. |
| resetBox | ResetBox | Choice | The box has been reset. |

**Table 191: RESTful Notification Events**

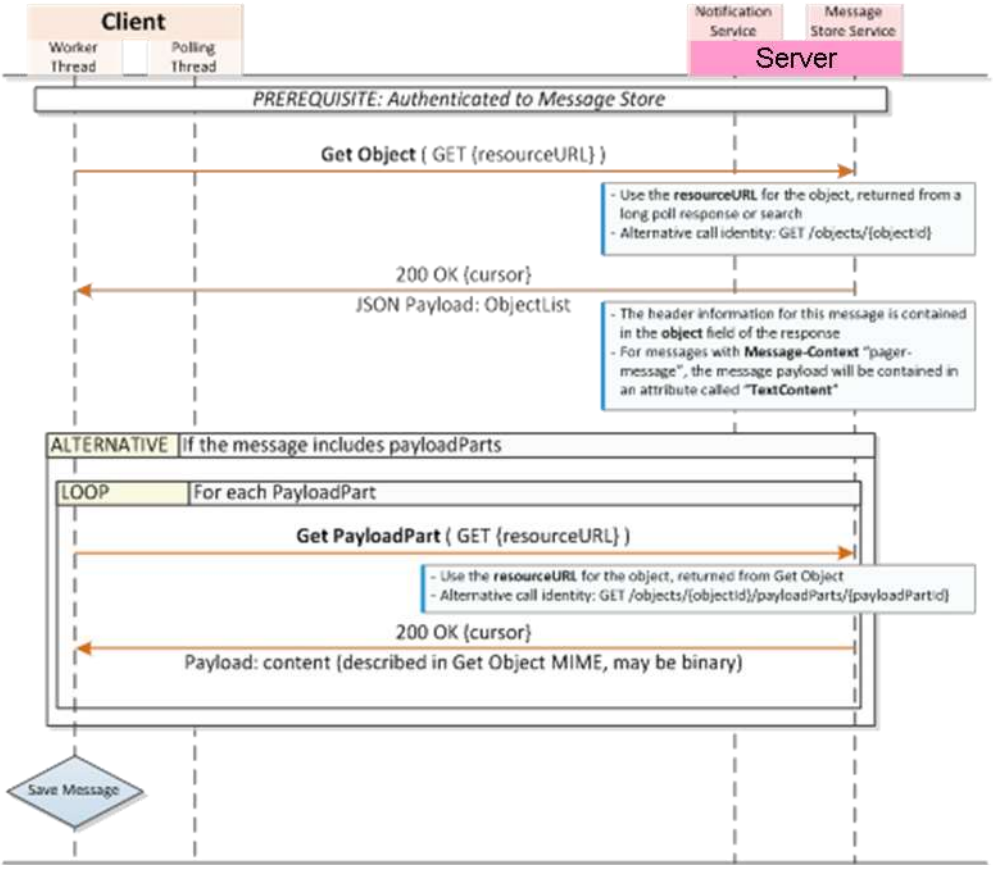
### B.2.3. Notification Channel Setup



### B.2.4. **Object Upload**




### B.2.5. **Example Object Download**



### B.2.6. Example RESTful Search operation

#### *B.2.6.1. /objects/operations/search*

Resource for getting information about a selected set of objects in the storage supporting POST only

`//{nmsHost}/nms/v1/base/{boxId}/objects/operations/search`

Note that NMS supports filtering and therefore a client may receive filtered results based on search criteria that the client sets or that the server imposes (based on client capabilities): searchCriteria, searchScope nor sortCriterion, NMS supports the maxEntries and fromCursor elements of selectionCriteria.

Objects are returned in descending date order; the most recent object is returned first.

##### *B.2.6.1.1. POST*

This endpoint is specified in more detail at [CPM-MSGSTOR-REST] section 6.7.5. The examples below do not exactly match the NMS specification because this version of NMS contains a restricted implementation of the search endpoint.

Making a request:

```
POST http://nms-
sib01.si.enclab.mno.net/nms/v1/base/tel:+19717774171/objects/operations/se
arch HTTP/1.1
Accept-Encoding: gzip,deflate
MIME-Version: 1.0
x-MNO-clientId: MNO_NMSG001
x-MNO-clientVersion: 2.3.4
x-MNO-deviceId: 310410438258561
x-MNO-contextInfo: mdl=MNONMSTestClient,os=1.2.3,fw=11.22.333.4444
x-MSw-ClientType: caching
Accept: application/json
Content-Type: application/json
Authorization: Bearer
PAT_bceNdO3GSKwkHBkL1rnEceW2TTLx3ijGzbonv3qFvVuYHnocDopEZ2eyMfzea/2POmQrLn
YJch2VDeEA2ooRh22pDnYEHk9YsFr8WrKbsMrHt5IMHhVrq81krtmWacbV/rFV4oT/1ckjpaIh
Yu+qh4YYLc3FGuYMl+fvipRoLBTYlyHUZSR7xJSVHqQtY1aIGU7J2xXr2r6tTIJ5eGpoRIHDK7
z3CL5c6ii1aI+TqyW5BBDqZtse8MTRfTva69ot9rv0ehYHoZ7T7Q/DfbY/DZ4ffnTReB6EHHYx
A62SOKJ+GiaAkVYZNhtuEAHTSAF6
Content-Length: 40
Host: nms-sib01.si.enclab.MNO.net
Proxy-Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
{"selectionCriteria": {"maxEntries": 3}}
```

Responses:

```
HTTP/1.1 200 OK
Server: nginx
Date: Thu, 22 Oct 2015 20:06:01 GMT
Content-Type: application/json; charset=UTF-8
Content-Length: 689
```

```
Cache-Control: proxy-revalidate
Proxy-Connection: Keep-Alive
Connection: Keep-Alive
Content-Encoding: gzip
Set-Cookie: SERVERID=882565574.20480.0000; path=/

{"objectList": {
   "object":    [
            {
         "parentFolder": "http://nms-
sib01.si.enclab.MNO.net/nms/v1/base/tel%3A%2B19717774171/folders/iw9%7CozA
EM333XXXMnMn",
         "attributes": {"attribute":          [
                        {
               "name": "Date",
               "value": ["2015-04-01T14:30:30Z"]
            },
                        {
               "name": "TextContent",
               "value": ["this is a test SMS message"]
            },
                        {
               "name": "Message-Context",
               "value": ["pager-message"]
            },
                        {
               "name": "Is-CPM-Group",
               "value": ["no"]
            },
                        {
               "name": "Direction",
               "value": ["IN"]
            },
                        {
               "name": "From",
               "value": ["+19995551212"]
            },
                        {
               "name": "Subject",
               "value": ["Test Object"]
            },
                        {
               "name": "To",
               "value": ["+19717774171"]
            }
         ]},
         "flags": {"flag": []},
         "path": "/Default/tel%3A%2B19995551212/CiUh",
         "resourceURL": "http://nms-
sib01.si.enclab.MNO.net/nms/v1/base/tel%3A%2B19717774171/objects/CiUh",
         "lastModSeq": 1445540857000000000,
         "correlationTag": "40bef8ea4cad717d"
```

```
      },
            {
         "parentFolder": "http://nms-
sib01.si.enclab.MNO.net/nms/v1/base/tel%3A%2B19717774171/folders/iw9%7CozA
EM333XXXMnMn",
         "attributes": {"attribute":          [
                        {
               "name": "Date",
               "value": ["2015-04-01T14:30:30Z"]
            },
                        {
               "name": "MultipartContentType",
               "value": ["application/vnd.wap.multipart.mixed"]
            },
                        {
               "name": "Message-Context",
               "value": ["standalone-message"]
            },
                        {
               "name": "Is-CPM-Group",
               "value": ["no"]
            },
                        {
               "name": "Direction",
               "value": ["IN"]
            },
                        {
               "name": "From",
               "value": ["+19995551212"]
            },
                        {
               "name": "Conversation-ID",
               "value": ["5EUINVAX8.CBHLZHDXR"]
            },
                        {
               "name": "Subject",
               "value": ["Test Object"]
            },
                        {
               "name": "To",
               "value": ["+19717774171"]
            }
         ]},
         "flags": {"flag": []},
         "path": "/Default/tel%3A%2B19995551212/wnhX",
         "resourceURL": "http://nms-
sib01.si.enclab.MNO.net/nms/v1/base/tel%3A%2B19717774171/objects/wnhX",
         "payloadPart":          [
                        {
               "contentType": "text/plain; name=sms",
               "size": 28,
               "href": "http://nms-
```

```
sib01.si.enclab.MNO.net/nms/v1/base/tel%3A%2B19717774171/objects/wnhX/payl
oadParts/M"
            },
                        {
               "contentType": "image/jpeg; name=testImage1.jpeg",
               "size": 130879,
               "href": "http://nms-
sib01.si.enclab.MNO.net/nms/v1/base/tel%3A%2B19717774171/objects/wnhX/payl
oadParts/n"
            }
         ],
         "lastModSeq": 0,
         "correlationId": "BNZ2HAFOV.70A9D1W5B"
      },
            {
         "parentFolder": "http://nms-
sib01.si.enclab.MNO.net/nms/v1/base/tel%3A%2B19717774171/folders/iw9%7CozA
EM333XXXMnMn",
         "attributes": {"attribute":          [
                        {
               "name": "Date",
               "value": ["2015-04-01T14:30:30Z"]
            },
                        {
               "name": "TextContent",
               "value": ["this is a test SMS message"]
            },
                        {
               "name": "Message-Context",
               "value": ["pager-message"]
            },
                        {
               "name": "Is-CPM-Group",
               "value": ["no"]
            },
                        {
               "name": "Direction",
               "value": ["IN"]
            },
                        {
               "name": "From",
               "value": ["+19995551212"]
            },
                        {
               "name": "Subject",
               "value": ["Test Object"]
            },
                        {
               "name": "To",
               "value": ["+19717774171"]
            }
         ]},
```

```
         "flags": {"flag":          [
            "\\Seen",
            "\\Flagged"
         ]},
         "path": "/Default/tel%3A%2B19995551212/CiU7",
         "resourceURL": "http://nms-
sib01.si.enclab.MNO.net/nms/v1/base/tel%3A%2B19717774171/objects/CiU7",
         "lastModSeq": 1445535875000000000,
         "correlationTag": "40bef8ea4cad717d"
      }
   ],
   "cursor": "wAnhuA$CAiUXA$TAhiA$xA3XA$"
}}
```

# Annex C Special Procedures

## C.1. SIP/TCP and NAT traversal

As specified in section 2.7 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 standardisation (see [IETF-DRAFT-RKEEP]). The procedures here will be updated once that work is completed. In particular this standardisation process should allow the client to detect that the network does not support network-initiated keep-alives as described above.

## C.2. 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.3. Definition of RCS CPIM Header Extensions

### C.3.1. RCS CPIM Extension Name Space

CPIM header extensions make use of the extension framework defined in [RFC3862]. The RCS extensions make use of the RCS name space as defined in this section.

The RCS Namespace is defined as follows:

```
NS: rcs <http://www.gsma.com>
```

NOTE: The namespace is considered as a place holder for a final one to be defined by GSMA or other committees.

### C.3.2. Definition of rcs.Service-Centre-Address header

The rcs.Service-Centre-Address header contains the Service Centre Address associated with a short message, see [3GPP TS 23.040]

The header is defined as an extension to the [RFC3862] field definitions. The limits for the occurrence of the field are defined in the following table:

| Field | Min Number | Max Number |
|---|---|---|
| rcs.Service-Centre-Address | 0 | 1 |

**Table 192: rcs.Service-Centre-Address header**

The field itself is defined in ABNF as follows:

```
service-centre-address  = "rcs.Service-Centre-Address:" service-
centre-value CRLF
service-centre-value    = "+" *15DIGIT
```

Example:

```
rcs.Service-Centre-Address: +491712020202
```

### C.3.3. Definition of rcs.Reply-Path header

The rcs.Reply-Path header contains the indication whether a reply path exists for a short message, see [3GPP TS 23.040].

The header is defined as an extension to the [RFC3862] field definitions. The limits for the occurrence of the field are defined in the following table:

| Field | Min Number | Max Number |
|---|---|---|
| rcs.Reply-Path | 0 | 1 |

**Table 193: rcs.Reply-Path header**

The field itself is defined in ABNF as follows:

```
reply-path  = "rcs.Reply-Path:" reply-path-value CRLF
reply-path-value  = "0" / "1"
```

Example:

```
rcs.Reply-Path: 0
```

### C.3.4. Definition of rcs.Replace-Short-Message-Type header

The rcs.Replace-Short-Message-Type header indicates the type number of a short message for replacement, see [3GPP TS 23.040]

The header is defined as an extension to the [RFC3862] field definitions. The limits for the occurrence of the field are defined in the following table:

| Field | Min Number | Max Number |
|---|---|---|
| rcs.Replace-Short-Message-Type | 0 | 1 |

**Table 194: Replace-Short-Message-Type header**

The field itself is defined in ABNF as follows:

```
rcs-replace-short-message-type = "rcs.Replace-Short-Message-Type:"
                  replace-type-value CRLF

replace-type-value      = "1" / "2" / "3" / "4" / "5" / "6" / "7"
```

Example:

```
rcs.Replace-Short-Message-Type: 1
```

### C.3.5. Definition of rcs.Mms-Message-Class header

The rcs.Mms-Message-Class header indicates the class of a multimedia message, see [3GPP TS 23.140].

The header is defined as an extension to the [RFC3862] field definitions. The limits for the occurrence of the field are defined in the following table:

| Field | Min Number | Max Number |
|---|---|---|
| rcs.Mms-Message-Class | 0 | 1 |

**Table 195: Mms-Message-Class header**

The field itself is defined in ABNF as follows:

```
rcs-mms-message-class = "rcs.Mms-Message-Class: "
                  class-value CRLF

class-value = "Personal" / "Advertisement" / "Informational" / "Auto"
```

Example:

```
rcs.Mms-Message-Class: Personal
```

### C.3.6. Definition of rcs.Message-Correlator header

The rcs.Message-Correlator header contains the message identification for message correlation, see section 3.2.4.7.2. The encoding of the message identification depends on the context defined in rcs.Message-Context parameter.

The header is defined as an extension to the [RFC3862] field definitions. The limits for the occurrence of the field are defined in the following table:

| Field | Min Number | Max Number |
|---|---|---|
| rcs.Message-Correlator | 0 | 1 |

**Table 196: Message-Correlator header**

The field itself is defined in ABNF as follows:

```
rcs-message-context = "rcs.Message-Correlator: "
                      message-correlator-value CRLF

                ; for encoding rules of "message-correlator-value"
                ; follow references in [CPM-MSGSTOR-REST]
```

Example:

```
rcs.Message-Correlator: Hello world
```

### C.3.7. Definition of rcs.Message-Context header

The rcs.Message-Context header indicates the context and presentation characteristics of a message. It can be used by interpreters to derive the encoding rules for the value of the rcs.Message-Context header.

The header is defined as an extension to the [RFC3862] field definitions. The limits for the occurrence of the field are defined in the following table:

| Field | Min Number | Max Number |
|---|---|---|
| rcs.Message-Context | 0 | 1 |

**Table 197: Message-Context header**

The field itself is defined in ABNF as follows:

```
rcs-message-context = "rcs.Message-Context: "
                      message-context-class CRLF

                ; for encoding rules of " message-context-class"
                ; refer to [RFC3458]
```

Example:

```
rcs.Message-Context: "pager-message"
```

## C.4. Definition of SIP Header Extensions

### C.4.1. User-Agent and Server Header Extensions

The "User-Agent" and "Server" header fields and their formatting are defined in [RFC3261]. For use in GSMA context, the header field values shall be composed in accordance with the definitions in this section.

The User-Agent and Server headers ABNF are specified in [RFC3261] 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 =  token [SLASH product-version]
```

```
product-version = token
```

The "User-Agent" or "Server" header value shall contain product tokens to identifiy the user agent characteristics.

For the use in GSMA context, the following ABNF applies:

```
Server = "Server" HCOLON product-list
User-Agent = "User-Agent" HCOLON product-list
product-list = enabler *(LWS enabler)
                    *(LWS service)
                    [LWS terminal]
                    [LWS client]
                    [LWS OS]
                    *(LWS list-extension)
```

The following user agent characteristics are defined:

1. Enabler technology:

   This product token identifies the supported and activated enabler technologies of an entity. The values of the product tokens are defined in the corresponding enabler specifications, e.g. the CPM enabler. One or more product tokens can be present in the header value. For RCS, the presence of the CPM technology product token defined in Annex D of [RCS-CPM-CONVFUNC-ENDORS] is mandatory. If used for other services, at least one enabler need to be specified as an extension of the enabler rule.

   ```
   enabler = CPM-Version / extension ; CPM version is defined in
                                     ; Annex D of
                                     ; [RCS-CPM-CONVFUNC-ENDORS]
   ```

2. Service:

   This product token identifies the supported and enabled services provided by an entity. Zero or more product tokens can be present in the header value. For RCS the service product token shall be provided by clients and may be provided by networks and Chatbot Platforms.

   ```
   service = RCS-product / extension
   RCS-product = "RCS-" RCS-device-token SLASH RCS-profile
   RCS-device-token = "client" ; header is included by a client
                    / "serv" ; header is included by an
                             ; application server
                    / "bot"  ; header is included by a botplatform
                    / token
   RCS-profile = token ; same as rcs_profile defined in section 2.3.2.2
   ```

   Where RCS-device-token shall be set to "client", "serv" and "bot" depending on whether the header field is included by a client, an application server or a Chatbot Platform respectively. RCS-profile shall be set to the same value as the one provided for the rcs_profile Configuration request parameter defined in section 2.3.2.2.

3. Terminal information:

This product token identifies the terminal on which an entity resides. The value of the product token is determined by the terminal OEM. Zero or one product token can be present in the header value. This product token is only applicable for clients.

```
terminal = "term-" terminal-vendor SLASH terminal-model "-"
            terminal-SW-version
terminal-vendor = token      ; same as terminal_vendor defined in
                             ; section 2.4 of [PRD-RCC.14]
terminal-model = sf-token ; same as terminal_model defined
                             ; in section 2.4 of [PRD-RCC.14]
terminal-SW-version = token  ; same as terminal_sw_version defined
                             ; in section 2.4 of [PRD-RCC.14]
sf-token = 1*(alphanum / "." / "!" / "%" / "*" / "_" /
                       "+" / "`" / "'" / "~" )
```

For RCS, the terminal-vendor, terminal-model and terminal-SW-version shall be set to the same value as the one provided for respectively the terminal_vendor, terminal_model and terminal_sw_version Configuration request parameters defined in section 2.4 of [PRD-RCC.14].

4. Client information:

   This product token identifies the client software. The value of the product token is determined by the client provider. Zero or one product token can be present in the header value. For RCS, the client information product token is only applicable for clients.

   ```
   client = "client-" client-vendor SLASH client-version
   client-vendor = token ; same as client_vendor defined in
                         ; section 2.3.2.2
   client-version = token; same as client_version defined in
                         ; section 2.3.2.2
   ```

   For RCS, the client-vendor and client-version shall be set to the same value as the one provided for respectively the client_vendor and client_version Configuration request parameters defined in section 2.3.2.2.

5. Operating System information:

   This product token identifies the operating System on which an entity resides. The value of the product token is determined by the terminal OS provider. Zero or one product token can be present in the header value. This product token is only applicable for clients.

   ```
   OS = "OS-" OS-type (SLASH OS-version)
   OS-type = "Android" / "IOS" / "SymbianOS" / "Windows" / "Other" /
              token
   OS-version = 1*alphanum *4("." 1*alphanum)
   ```

   Where OS-type and OS-version should be set to the most appropriate values for the Operating System on which the client is running.

6. Vendor specific and extensions:

The extensibility of the product token list is maintained. Vendor specific information and extensions beyond the defined product token groups can be included in the list using the extension mechanism.

```
list-extension = extension
extension = extension-product / comment
extension-product = gsma-extension / other-extension
gsma-extension = tag "-" value [SLASH product-version]
tag = sf-token
value = token
product-version = token
other-extension = token [SLASH product-version]
```

NOTE: Because it cannot be guaranteed that the User-Agent and Server header fields are passed unmodified between entities supporting RCS, it is recommended that such entities do not depend on the presence of the described information. It is therefore recommended to handle this as purely informational.

Examples:

```
User-Agent: CPM-client/OMA2.2 RCS-client/UP_2.0 term-Vendor1/Model1-
XXXX client-CLN1/Software1234 OS-Android/8.1
Server: CPM-serv/OMA2.2 RCS-serv/UP_2.0
```

# 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<br>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 |
| 6.0 | 28 February 2015 | First version of the document for RCS 5.3: Include approved CR1005 | PSMC | Tom Van Pelt / GSMA |
| 7.0 | 21 March 2016 | First version of the document for RCS 6.0: Include approved CR1007 | PSMC | Tom Van Pelt / GSMA |
| 8.0 | 28 June 2017 | First version of the document for RCS 7.0: Include approved CR1008 | TG | Tom Van Pelt / GSMA |
| 9.0 | 16 May 2018 | Include approved CR1009 | TG | Tom Van Pelt / GSMA |
| 10.0 | 06 December 2018 | Include approved CR1002 | TG | Tom Van Pelt / GSMA |

## Other Information

| Type | Description |
|---|---|
| Document Owner | Network Group, 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.