NNI Test Case Specification

Version 1.0

25 September 2013

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

Security Classification: Non-confidential

Access to and distribution of this document is restricted to the persons permitted by the security classification. This document is confidential to the Association and is subject to copyright protection. This document is to be used only for the purposes for which it has been supplied and information contained in it must not be disclosed or in any other way made available, in whole or in part, to persons other than those permitted under the security classification without the prior written approval of the Association.

Copyright Notice

Disclaimer

The GSM Association ("Association") makes no representation, warranty or undertaking (express or implied) with respect to and does not accept any responsibility for, and hereby disclaims liability for the accuracy or completeness or timeliness of the information contained in this document. The information contained in this document may be subject to change without prior notice.

Antitrust Notice

The information contain herein is in full compliance with the GSM Association's antitrust compliance policy.

1. Introduction

1.1 General

This document is intended for use as test specification for IR.90 RCS Interworking Guidelines.

The test cases in this document are partitioned to correspond to chapters in IR.90.

1.2 Functional assumptions

Users A, B, C and D belong to different operators/service providers

1. All users have been provisioned in their respective networks.
2. All users’ telephones have been configured unless otherwise specified.
3. All users are registered in their RCS networks unless otherwise specified. The registration timer for each user is far from expiration.
4. All users are in 3G/LTE mobile coverage unless otherwise specified. Wi-Fi coverage is specified explicitly.
5. Operators/service providers that are performing testing have signed the appropriate interworking agreements and have IMS level of interconnections.

1.3 Testing Environment Assumptions

All users’ operators have successfully verified RCS implementations, including UNI, in their respective networks.

ENUMs and DNSes have been established and provisioned, if applicable.

All users use UEs of their operator’s choice.

Repeat the test swapping the role of the devices.

Tests are performed as applicable, by subscriber and terminal capabilities.

It is assumed that the following testing over NNI has been concluded prior to execution of RCS 5.1 NNI Test Cases (see also RCS 5.1 NNI spreadsheet, Annex A to IR.90):

1. IP connectivity
2. SIP connectivity
3. RCS media connectivity
4. Addressing and routing, including ENUM and DNS functionality

1.4 Assumed architecture

Reference to IR.65 for IMS Interconnect architecture

1.5 References

Document Number	Title
[3GPP TS 26.114]	IP Multimedia Subsystem (IMS); Multimedia Telephony; Media handling and interaction
[3GPP TS 24.173]	IMS Multimedia telephony communication service and supplementary services; Stage 3
[3GPP TS 24.229]	IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3
[3GPP TS 24.610]	Communication HOLD (HOLD) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification
[ITU H.264]	Advanced video coding for generic audiovisual services
[GSMA IR.65]	IMS Roaming and Interworking Guidelines v12.0
[GSMA IR.74]	Video Share Interoperability Specification v1.4
[GSMA IR.79]	Image Share Interoperability Specification v1.4
[GSMA IR.90]	RCS Interworking Guidelines v5.0
[GSMA RCS 5.1]	Rich Communication Suite 5.1 Advanced Communications, Services and Client Specification v3.0

1.6 Acronyms

Acronym / Abbreviation	Description
3GPP	3rd Generation Partnership Project
AMR	Adaptive Multi-Rate
AVP	Audio Video Profile
BPEF	Blacklist Policy Enforcement Function
CBP	Constrained Baseline Profile
CPIM	Common Profile for Instant Messaging
CPM	Converged IP Messaging
CS	Circuit Switched
DNS	Domain Name System
EAB	Enhanced Address Book
ENUM	E.164 Number Mapping
FT	File Transfer
GC	Group Chat
GRUU	Globally Routable User agent URI

GSMA	GSM Association
HSPA	High Speed Packet Access
HTTP	Hyper-Text Transfer Protocol
HTTPS	Hyper-Text Transfer Protocol Secure
IM	Instant Messaging
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPX	Internet Protocol Packet Exchange
LTE	Long Term Evolution
MIME	Multipurpose Internet Mail Extensions
MMC	Mobile Country Code
MNC	Mobile Network Code
MNO	Mobile Network Operator
MSRP	Message Session Relay Protocol
NAT	Network Address Translation
NB AMR	Narrowband AMR
NNI	Network-to-Network Interface
OMA	Open Mobile Alliance
PNB	Personal Network Blacklist
RCS	Rich Communication Suite
RCS-AA	RCS Access Agnostic mode
RCS-CS	RCS CS mode
RR	Receiver Report
RTCP	RTP Control Protocol
RTP	Real Time Protocol
S&F	Store and Forward
SDP	Session Description Protocol
SIMPLE	Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions
SIP	Session Initiation Protocol
SPI	Social Presence Information
UE	User Equipment
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UX	User Experience

VIP	Very Important Person
VoHSPA	Voice over HSPA
VoLTE	Voice over LTE
WB AMR	Wideband AMR
XDMS	XML Document Management Server
XML	eXtensible Markup Language

2 Introduction and Overall IMS NNI Architecture

2.1 Configuration and registration

No tests

2.2 Roaming

No tests

2.3 Legacy

No tests

3 Capability Discovery

3.1 SIP OPTIONS based

3.1.1 Pre-call capability discovery

Test case ID	**RCS_3_1_1_1**
Related Test Cases	
Feature	Capability Discovery - SIP OPTIONS based
Purpose	Initial check of service capabilities of RCS users.
Pre-conditions Scenario	A and B are RCS users and online.

Test procedure	A initiates Capability Discovery with B.
Expected results **Post-conditions**	A’s device shows B’s correct service capabilities. B’s device shows A’s correct service capabilities.
Deep inspection	Verify SIP OPTIONS exchange with correct service capability tags of A and B.

Test case ID	**RCS_3_1_1_2**
Related test cases	**RCS_3_1_1_3**
Feature	Capability Discovery - SIP OPTIONS based
Purpose	Capability Discovery with non-IMS enabled, non-RCS user.
Pre-conditions Scenario	A is an RCS user. B is non-IMS enabled and non-RCS user.
Test procedure	A initiates Capability Discovery with B.
Expected results **Post-conditions**	B’s service capabilities displayed in A’s device remain unchanged.
Deep inspection	Verify 404 NOT FOUND response is returned from B’s network.

Test case ID	**RCS_3_1_1_3**
Related test cases	
Feature	Capability Discovery - SIP OPTIONS based
Purpose	Capability Discovery with IMS enabled non-RCS user.
Pre-conditions Scenario	B is IMS enabled and non-RCS user.
Test procedure	A initiates Capability Discovery with B.
Expected results **Post-conditions**	No service capabilities from B are displayed in A's device.
Deep inspection	Verify that no GSMA RCS 5.1 [GSMA RCC.07] service tags are returned from B's network.

Test case ID	**RCS_3_1_1_4**
Related test cases	
Feature	Capability Discovery - SIP OPTIONS based
Purpose	Capability exchange with RCS-capable B, which is not registered in IMS.
Pre-conditions Scenario	1. B is an RCS user who is currently not registered in IMS. 2. IM CAP ALWAYS ON and FT CAP ALWAYS ON are not enabled on A.

Test procedure	A initiates Capability Discovery with B.
Expected results **Post-conditions**	No service capabilities from B are displayed in A’s device.
Deep inspection	Verify that 480 TEMPORARILY UNAVAILABLE or 408 REQUEST TIMEOUT or 487 REQUEST TERMINATED is returned from B’s network.

Test case ID	**RCS_3_1_1_5**
Related test cases	
Feature	Capability Discovery - SIP OPTIONS based
Purpose	Capability exchange with RCS-capable B, which has just lost connectivity but remains registered in IMS (e.g., suddenly removing battery from B’s device).
Pre-conditions Scenario	1. B loses connectivity but remains registered in IMS (e.g., suddenly removing battery from B’s device). 2. IM CAP ALWAYS ON and FT CAP ALWAYS ON are not enabled on A.
Test procedure	1. A selects B. 2. Capability exchange takes place.
Expected results **Post-conditions**	B’s capabilities are not available to A since the network reports that B is offline.
Deep inspection	1. Verify SIP OPTIONS exchange. 2. Verify that the response from the IMS core is 408 (REQUEST TIMEOUT) or 487 (REQUEST TERMINATED).

3.1.2 In-call/in-session capability discovery

No tests

3.1.3 Multi-device handling

No tests

3.1.4 Exception conditions

Test case ID	**RCS_3_1_4_1**
Related test cases	
Feature	Capability Discovery - SIP OPTIONS based
Purpose	Capability query in 1-1 chat (file transfer no longer available due to 2G coverage).
Pre-conditions Scenario	1. A and B are engaged in a 1-1 chat. 2. FT CAP ALWAYS ON is not set on A.
Test procedure	1. B moves to 2G coverage. 2. That triggers a capability exchange started from B to update A about the fact that file transfer (FT) is no longer supported.
Expected results **Post-conditions**	FT is not supported by B in A's device.
Deep inspection	Verify OPTIONS exchange with correct service capability tags from B to A.

Test case ID	**RCS_3_1_4_2**
Related test cases	

Feature	Capability Discovery - SIP OPTIONS based
Purpose	Capability query in 1-1 chat (FT becomes available)
Pre-conditions Scenario	1. B is under 2G coverage. 2. A and B are in an RCS 1-1 chat. 3. FT CAP ALWAYS ON is not set on A.
Test procedure	1. B moves to 3G/HSPA coverage 2. That triggers a capability exchange started from B to update A about the fact that FT is supported.
Expected results **Post-conditions**	A’s device shows that B supports FT.
Deep inspection	1. Verify OPTIONS exchange with correct service capability tags from B to A. 2. Verify response from the core is 200 OK.

3.2 *Presence based*

3.2.1 *User Discovery*

Test case ID	**RCS_3_2_1_1**
Related Test Cases	
Feature	Capability Discovery - Presence Based
Purpose	User discovery: Device first-time registration
Pre-conditions Scenario	1. CAPABILITY DISCOVERY MECHANISM = PRESENCE 2. Optional CAPABILITY POLLING is supported by the operator. 3. B and C are contacts in A’s Enhanced Address Book (EAB); B and C are registered and available. 4. A has not yet initially registered in her service provider’s network.

Test procedure	A performs first-time registration and configuration.
Expected results **Post-conditions**	After some period of time, A's EAB correctly displays that B and C are RCS contacts.
Deep inspection	1. Anonymous SUBSCRIBE is seen from A to B and A to C 2. The resulting NOTIFY messages at the NNI contain the appropriate service capabilities for B and C.

Test case ID	**RCS_3_2_1_2**
Related Test Cases	**RCS_3_2_1_1**
Feature	Capability Discovery - Presence Based
Purpose	User discovery: Registered non-RCS contact added to EAB.
Pre-conditions Scenario	1. B is a non-RCS user and not in A's EAB and registered in IMS. 2. B is not presence enabled.
Test procedure	A adds B in A's EAB.
Expected results **Post-conditions**	A's EAB indicates that B is not an RCS user.
Deep inspection	1. In the case where there is IMS peering between carriers for services other than RCS, an anonymous SUBSCRIBE may be sent over the NNI, but it is expected that an error code of some sort would be returned; there would normally not be a NOTIFY. 2. If B is in a network with a presence server, it is likely that if a NOTIFY is returned it would not contain any RCS service capabilities. It is possible, but

	not likely, that the remote non-RCS domain would support anonymous operations.

Test case ID	**RCS_3_2_1_3**
Related Test Cases	**RCS_3_2_1_1**
Feature	Capability Discovery - Presence Based
Purpose	User discovery: Not registered RCS contact added to EAB
Pre-conditions Scenario	B is a provisioned RCS user and not in A's EAB and not registered in IMS (e.g., all devices powered off).
Test procedure	A adds B in A's EAB.
Expected results **Post-conditions**	1. A's EAB indicates that B is an RCS user. 2. A's EAB indicates that B's has no specific service capabilities available.
Deep inspection	1. Anonymous SUBSCRIBE is seen over the NNI from A for B. 2. The resulting NOTIFY message to A at the NNI contains no service capabilities because user B has no active devices registered.

3.2.2 Capability Update

Test case ID	**RCS_3_2_2_1**
Related Test Cases	
Feature	Capability Discovery - Presence Based
Purpose	Capability update: User interacts with RCS contact.

Pre-conditions Scenario	A and B are registered RCS users.
Test procedure	1. B interacts with A. 2. B's device performs a capability update fetch. 3. A's capability information is updated on B's device.
Expected results **Post-conditions**	A's current capability information is correctly delivered to B.
Deep inspection	1. Device B sends an anonymous SUBSCRIBE over NNI to A. 2. The resulting NOTIFY routed back to B accurately characterizes the service capabilities based on operator service and network connection status, policy, and device status.

Test case ID	**RCS_3_2_2_2**
Related Test Cases	
Feature	Capability Discovery - Presence Based
Purpose	Operator has a service availability policy that is contingent upon device network / attachment status.
Pre-conditions Scenario	1. Operator has a service availability policy that is contingent upon network connectivity type and status. 2. Device B implements such a policy for a specific service for presence-based capability discovery/capability update.
Test procedure	1. Device B is in a network coverage / attachment state that, per operator policy and device capabilities, publishes that one or more services are available. The device may also publish service descriptions that are provided independent of network coverage / attachment state. 2. A performs capability update fetch. 3. B's capability changes (i.e., B changes network coverage / attachment status such that one or more services are affected). 4. A performs capability update fetch.

Expected results **Post-conditions**	1. For the first fetch, A’s capability information indicates the set of network-dependent services that properly reflect device B’s status and operator policy. 2. For the second fetch, A’s network attachment status has changed and one or more services are affected; thus A’s capability information now indicates the changed set of services that properly reflect device B’s status and operator policy.
Deep inspection	1. Each fetch results in a NOTIFY being generated that accurately characterizes the service availability for each user based on operator service and network connection status policy and device status. 2. The cause of the differences in the service information provided between the first and second fetch is that between the two fetches: a. Device B network/attachment status changed. b. Device B published the changed set of services to its Presence Server. c. Note that neither of these events is visible at the NNI, but unless they both occur, no change will be detected between the two fetches.

3.2.3 Multi-device handling

Test case ID	**RCS_3_2_3_1**
Related Test Cases	**RCS_3_2_1_1**
Feature	Capability Discovery - Presence Based
Purpose	User Discovery: Registered RCS contact with multiple registered devices added to EAB
Pre-conditions Scenario	1. CAPABILITY DISCOVERY MECHANISM = PRESENCE. 2. B is a provisioned RCS user, not in A’s EAB, and is registered in IMS. 3. B has multiple RCS devices registered and active in the network.
Test procedure	A adds B in A’s EAB.
Expected results **Post-conditions**	B’s aggregated service capabilities are delivered to A
Deep inspection	1. Anonymous SUBSCRIBE is seen at NNI from A to B. 2. The resulting NOTIFY message at the NNI contains the aggregated service capabilities of B.

3.2.4 Feature Interaction

Test case ID	**RCS_3_2_4_1**
Related Test Cases	**RCS_5_1_1, RCS_5_2_1, RCS_5_2_2**
Feature	Capability Discovery - Presence Based
Purpose	User Capability and Social Presence Information (SPI) service interaction
Pre-conditions Scenario	1. Social Presence is enabled via Presence Server (PRESENCE PROFILE = 1). 2. A is sharing SPI with B. (**RCS_5_1_1**).
Test procedure	B interacts with A. This could be in any form including an incoming call, messaging interface, or interaction at the EAB.
Expected results **Post-conditions**	Anonymous fetches are not seen across the NNI for SPI contacts.
Deep inspection	1. B's device does not perform capability update fetches. 2. If A is a VIP buddy, then B immediately and always sees A's current capability information. Changes in SPI information status are sent automatically to B and are available for display. (**RCS_5_2_1**) 3. If B is a non-VIP buddy then a non VIP SPI fetch will be performed by B's device. (**RCS_5_2_2**)

Test case ID	**RCS_3_2_4_2**
Related Test Cases	**RCS_5_1_1**
Feature	Capability Discovery - Presence Based
Purpose	SPI Who Can I Invite? RCS contact / capability discover interaction
Pre-conditions Scenario	1. Social Presence is enabled via Presence Server (PRESENCE PROFILE = 1). 2. A is not sharing SPI with B. 3. B is RCS and SPI sharing capable.

Test procedure	1. A interacts with contact B. 2. A’s EAB shows B can be invited to share RCS SPI. 3. A may now choose to invite B to share RCS SPI (i.e., make B an RCS buddy).
Expected results **Post-conditions**	A can see that B supports SPI.
Deep inspection	1. A sends an anonymous SUB SCRIBE over NNI for B. 2. The resulting NOTIFY contains the feature tag for SPI.

Test case ID	**RCS_3_2_4_3**
Related Test Cases	
Feature	Capability Discovery - Presence Based
Purpose	SIP OPTIONS and Presence based capabilities interaction based on common device stack client (as defined 2.6.1.3.2 of RCC.07)
Pre-conditions Scenario	1. CAPABILITY DISCOVERY MECHANISM = PRESENCE. 2. Optional CAPABILITY POLLING is supported by the operator A. 3. CAPABILITY DISCOVERY VIA COMMON STACK =1. 4. B is a contact in A’s EAB. B is in a network that does not support presence-based capabilities discovery (only SIP OPTIONS supported). 5. A has not yet initially registered in A’s service provider’s network.
Test procedure	A performs first-time registration and configuration.
Expected results **Post-conditions**	B’s service capabilities is available to A.
Deep inspection	1. Anonymous SUBSCRIBE is seen over the NNI for contact B. 2. Contact B’s network responds to the SUBSCRIBE with a 405 or a 501. 3. The client A understands that these specific codes indicate that the specified contact requires SIP OPTIONS. 4. The client A performs a SIP OPTIONS exchange and displays the results to User A.

	Note: This will not work for an anonymous SUBSCRIBE that was generated by an RLS unless the PS and device have defined a specific RLMI resource state that uniquely identifies list members for which these specific error codes have been received. These values should probably be defined and normalized in the RCS UNI spec.

3.2.5 Exception conditions

Test case ID	**RCS_3_2_5_1**
Related Test Cases	**RCS_3_2_1 _1**
Feature	Capability Discovery - Presence Based
Purpose	User Capability: RCS feature tag recognized, not supported.
Pre-conditions Scenario	B's service provider supports an RCS feature that is not supported by A's service provider.
Test procedure	A performs an action triggering an update of A's contact's user capability information (e.g., she interacts with the contact).
Expected results **Post-conditions**	B's capability information is delivered to A.
Deep inspection	1. A generates an anonymous SUBSCRIBE over the NNI to B. 2. The resulting NOTIFY from B accurately characterizes the service capabilities based on operator service and network connection status policy and device status. 3. If both operators use the same RCS services, or if any service filtering mechanism is used to strip such descriptions from the NOTIFY before it is routed over the NNI, then this is not testable.

4 IP Interconnection

No tests (see General Assumptions)

5 Social Presence

5.1 *Buddy List Management*

Test case ID	**RCS_5_1_1**
Related Test Cases	**RCS_5_1_2, RCS_5_1_3, RCS_5_1_4, RCS_5_1_5, RCS_5_1_6, RCS_3_2_4_2**
Feature	Social Presence - Buddy List Management
Purpose	Social Presence: Sharing invitation accept
Pre-conditions Scenario	1. Social Presence is enabled for both A and B via Presence Server (PRESENCE PROFILE = 1). 2. Both devices have an active buddy list. 3. A and B are not in each other's buddy lists (they are not already buddies). 4. B has not blocked A. 5. A can see that B supports SPI (Who can I invite – test **RCS_3_2_4_2**).
Test procedure	1. A invites B to share SPI. 2. B receives the SPI sharing invitation and accepts.
Expected results **Post-conditions**	A and B can see each other's SPI.
Deep inspection	1. A SIP SUBSCRIBE will be sent from A to B over NNI. 2. The response to SUBSCRIBE is 202 Accepted. This will be followed by a "dummy" NOTIFY with a pending subscription state. 3. After B accepts the invite: a. There will be a full NOTIFY from B to A with an active subscription state, this NOTIFY will contain USER B's SPI info. b. B network will send a SUBSCRIBE over NNI which will result in a 200 OK and NOTIFY for active subscriptions with A's SPI info.

Test case ID	**RCS_5_1_2**
Related Test Cases	**RCS_5_1_1, RCS_5_1_3, RCS_5_1_4, RCS_5_1_5, RCS_5_1_6**
Feature	Social Presence - Buddy List Management
Purpose	Social Presence: Sharing invitation ignore

Pre-conditions Scenario	1. Social Presence is enabled for both A and B via Presence Server (PRESENCE PROFILE = 1). 2. Both devices have active buddy list and winfo subscriptions. 3. A and B are not in each other's buddy lists (they are not already buddies). 4. B has not blocked A.
Test procedure	1. A invites B to share SPI. 2. B receives the SPI sharing invitation and ignores it. 3. A sees state as "pending" until B accepts, or blocks or a timeout occurs.
Expected results **Post-conditions**	1. B receives an indication that the invitation has not been accepted. 2. A and B are not able to see each other's SPI.
Deep inspection	1. A SIP SUBSCRIBE will be sent from A to B over NNI. 2. Response to SUBSCRIBE is 202 Accepted. This will be followed by a "dummy" NOTIFY with a pending subscription state.

Test case ID	**RCS_5_1_3**
Related Test Cases	**RCS_5_1_1, RCS_5_1_2, RCS_5_1_4, RCS_5_1_5, RCS_5_1_6**
Feature	Social Presence - Buddy List Management
Purpose	Social Presence: Sharing invitation blocked.
Pre-conditions Scenario	1. Social Presence is enabled for both A and B via Presence Server (PRESENCE PROFILE = 1). 2. Both devices have active buddy list and winfo subscriptions. 3. A and B are not in each other's buddy lists (they are not already buddies). 4. B has not blocked A.
Test procedure	1. A invites B to share SPI. 2. B blocks the SPI sharing invitation. 3. A again invites B to share SPI. 4. B does not get another sharing invitation (User A is in blocked-contacts).
Expected results **Post-conditions**	1. A does receive an indication that the invitation was being blocked (status terminated/rejected). 2. B does not receive A's SPI sharing invitations subsequent to blocking B. 3. A and B are not able to see each other's SPI. 4. B does not see any subsequent invitations from B to share SPI.

Deep inspection	1. A SIP SUBSCRIBE will be sent from A to B over NNI. 2. The response to SUBSCRIBE is 202 Accepted. This will be followed by a “dummy” NOTIFY with a pending subscription state. 3. After User B blocks the invite, there will be a NOTIFY from B to A with a terminated, rejected subscription state. This NOTIFY will not contain B’s SPI. 4. Any subsequent SUBSCRIBE from A will result in a NOTIFY with a terminated, rejected subscription state.

Test case ID	**RCS_5_1_4**
Related Test Cases	**RCS_5_1_1, RCS_5_1_2, RCS_5_1_3, RCS_5_1_5, RCS_5_1_6**
Feature	Social Presence - Buddy List Management
Purpose	Social Presence: Sharing invitation unblocked
Pre-conditions Scenario	1. Social Presence is enabled via Presence Server (PRESENCE PROFILE = 1). 2. Both devices have active buddy list and winfo subscriptions. 3. B has blocked A’s SPIs (test **RCS_5_1_1**) 4. B is not currently in A’s buddy list.
Test procedure	1. B unblocks A. 2. A invites B to share SPI. 3. B receives the SPI sharing invitation and accepts.
Expected results **Post-conditions**	1. Subsequent to the B accepting A’s buddy request: 2. A and B can see each other’s SPI.
Deep inspection	Deep inspection is same for the accept case as defined in test **RCS_5_1_1**.

Test case ID	**RCS_5_1_5**
Related Test Cases	**RCS_5_1_1, RCS_5_1_2, RCS_5_1_3, RCS_5_1_4, RCS_5_1_6**
Feature	Social Presence - Buddy List Management
Purpose	Social Presence: Sharing revoked

Pre-conditions Scenario	1. Social Presence is enabled via Presence Server (PRESENCE PROFILE = 1). 2. Both devices have active buddy list and winfo subscriptions. 3. A and B are sharing SPI (test **RCS_5_1_1**). 4. A and B can each see each other's SPI.
Test procedure	A revokes SPI sharing with B.
Expected results **Post-conditions**	1. A no longer sees B's SPI in the EAB. 2. B no longer sees A's SPI in the EAB. 3. B can see that the relationship was revoked. 4. B's contact information (e.g., vCard) remains in A's address book.
Deep inspection	1. Subsequent to A placing B in the revoked contact list, B's network will issue a SIP SUBSCRIBE for B that has an "expires" time of "0" to close the subscription. B's network will respond with a NOTIFY with a subscription state of "terminated". 2. A's network will issue a NOTIFY to the existing SUBSCRIBE from B to A with a subscription state of terminated, rejected. 3. As a result, the active subscriptions in both directions have been terminated.

Test case ID	**RCS_5_1_6**
Related Test Cases	**RCS_5_1_1, RCS_5_1_2, RCS_5_1_3**
Feature	Social Presence - Buddy List Management
Purpose	Social Presence: Personalized invitation
Pre-conditions Scenario	1. Social Presence is enabled via Presence Server (PRESENCE PROFILE = 1). 2. Both devices have active buddy list and winfo subscriptions. 3. A is not a contact in B's EAB. 4. A and B are not in each other's buddy lists (they are not already buddies). 5. B has not blocked A.
Test procedure	1. A defines a nickname at A's UE. 2. A sends an invitation (with the nickname) to B to share SPI from that UE.

Expected results **Post-conditions**	B receives the SPI sharing invitation. Since A is not defined in B's local address book, the initiation to share SPI should use the nickname defined by A (this is not visible at NNI but validates that B received it and UE uses it properly).
Deep inspection	A's defined nickname is included as part of the identity information provided in the "P-Preferred-Identity" and "From" header of the SIP SUBSCRIBE request from A to B.

5.2 *VIP/non-VIP*

Test case ID	**RCS_5_2_1**
Related Test Cases	**RCS_5_1_1**
Feature	Social Presence - VIP Subscription over NNI.
Purpose	Social Presence: Successful sharing of SPI over NNI between VIP contacts.
Pre-conditions Scenario	1. A and B are sharing SPI. (test **RCS_5_1_1**) 2. A and B's devices are on the network, initialized, and have active subscriptions to their VIP lists. 3. Operator policy and device configuration supports refreshing VIP subscriptions.
Test procedure	1. A and B can see each other's SPI. 2. Each user makes changes to one or more SPI attributes. 3. The expires time used for the VIP subscription expires. 4. Each user makes changes to one or more SPI attributes.
Expected results **Post-conditions**	1. A and B can see the updates to SPI information. 2. After the expires time, A and B can see updates to SPI information.
Deep inspection	1. VIP list information is propagated using long lived subscriptions. At the NNI, individual long lived subscriptions should be seen for each watcher/presentity. 2. When any user makes changes to SPI information, a NOTIFY should be seen at the NNI which contains the latest SPI information. 3. Watchers/RLS should refresh subscriptions before they expire so that users do not detect an interruption in VIP SPI service. 4. Thus, prior to any individual SUBSCRIBE reaching its expires time, the RLS should send a refresh SUBSCRIBE over the NNI to maintain the subscription in the active state.

	5. This does not specify variations that may be seen contingent upon any optimizations that may be in use over the NNI, including condition notifications, filtering, notification suppression, or throttling.

Test case ID	**RCS_5_2_2**
Related Test Cases	**RCS_5_1_1**
Feature	Social Presence - non-VIP Subscription over NNI.
Purpose	Social Presence: Successful sharing of SPI over NNI between non-VIP contacts.
Pre-conditions Scenario	1. A and B are sharing SPI. (test **RCS_5_1_1**) 2. A and B are non-VIP contacts with each other.
Test procedure	A interacts with B on the local EAB.
Expected results **Post-conditions**	A can see B’s SPI information.
Deep inspection	1. At the NNI, individual anonymous fetch subscription should be seen from A to B. 2. A NOTIFY will be sent from B to A over the NNI; the NOTIFY contains B’s SPI.

5.3 *SPI Attribute Management*

Test case ID	**RCS_5_3_1**
Related Test Cases	**RCS_5_1_1, RCS_5_2_1, RCS_5_2_2**
Feature	Social Presence - SPI attribute management
Purpose	Social Presence: Availability changed from unavailable to available

Pre-conditions Scenario	1. Social Presence is enabled via Presence Server (PRESENCE PROFILE = 1). 2. Availability is enabled (AVAILABILITY AUTHORIZATION = 1). 3. A and B are sharing SPI. (test **RCS_5_1_1**) 4. A and B are VIP contacts. 5. A has defined A's availability status to "unavailable".
Test procedure	A changes A's availability from Unavailable to Available.
Expected results **Post-conditions**	B sees A's status as Available.
Deep inspection	When A changes availability status, a NOTIFY is sent from A to B over NNI that contains the new information.

Test case ID	**RCS_5_3_2**
Related Test Cases	**RCS_5_1_1, RCS_5_2_1, RCS_5_2_2**
Feature	Social Presence - SPI attribute management
Purpose	Social Presence: Portrait icon change
Pre-conditions Scenario	1. Social Presence is enabled via Presence Server (PRESENCE PROFILE = 1). 2. A and B are sharing SPI. (test **RCS_5_1_1**). 3. A and B are VIP contacts. 4. A has defined a portrait icon.
Test procedure	A changes its portrait icon.
Expected results **Post-conditions**	B sees A's new portrait icon.

Deep inspection	1. When A changes its portrait icon, a NOTIFY is sent from A to B over NNI that contains the link to the icon which resides in the Presence Content XDMS of A's operator. 2. B's device must detect the change of link, and perform XCAP GET to retrieve the updated icon via a transaction with local domain XCAP Root URI. This is not visible at the NNI. 3. The inter-network Proxy of User B will forward the HTTP GET from B over the HTTP NNI point that is addressed to the inter-network proxy of User A's XDMS. 4. The GET at the NNI will result in a 200 OK that contains the updated icon as a MIME attachment which will be forwarded to B for display.

Test case ID	**RCS_5_3_3**
Related Test Cases	**RCS_5_1_1, RCS_5_2_1, RCS_5_2_2**
Feature	Social Presence - SPI attribute management
Purpose	Social Presence: Free Text change
Pre-conditions Scenario	1. Social Presence is enabled via Presence Server (PRESENCE PROFILE = 1). 2. A and B are sharing SPI. (test **RCS_5_1_1**) 3. A and B are VIP contacts. 4. A has defined Free Text.
Test procedure	A changes A's Free Text.
Expected results **Post-conditions**	B sees A's new free text.
Deep inspection	When A changes its free text, a NOTIFY is sent from A to B over NNI that contains the new information.

Test case ID	**RCS_5_3_4**
Related Test Cases	**RCS_5_1_1, RCS_5_2_1, RCS_5_2_2**
Feature	Social Presence - SPI attribute management

Purpose	Social Presence: Favourite link and label change
Pre-conditions Scenario	1. Social Presence is enabled via Presence Server (PRESENCE PROFILE = 1). 2. A and B are sharing FPI. (test **RCS_5_1_1**) 3. A and B are VIP contacts. 4. A has defined Favourite link and label.
Test procedure	A sets/changes A's Favourite link and label information.
Expected results **Post-conditions**	B sees A's updated Favourite link and label information.
Deep inspection	When A changes the Favourite link and label info, a NOTIFY is sent from A to B over NNI that contains the new information. Note / Example values: A wants to share the blog with friends. A fills in the URL with www.userA_blog.com and fills in the label with "Pizza Party Photos".

5.4 *Location Management*

Test case ID	**RCS_5_4_1**
Related Test Cases	**RCS_5_1_1, RCS_5_2_1, RCS_5_2_2**
Feature	Social Presence - Location management
Purpose	Geolocation Declarative text and location sharing policy management; authorize and revoke
Pre-conditions Scenario	1. Social Presence is enabled via Presence Server (PRESENCE PROFILE = 1). 2. A is sharing SPI with B and C. (test **RCS_5_1_1**) 3. B and C are in the same network. 4. A, B, and C are VIP contacts. 5. None of the users has authorized location sharing.
Test procedure	1. A sets A's declarative text location manually (e.g., "I'm in Austin, Texas"). 2. A enables location sharing with B. 3. After some period of time (after verifying B has seen location information), A revokes location sharing information with B.

Expected results **Post-conditions**	1. A’s location information is not visible to B or C. 2. After A enables location sharing with B, B can see the declarative text information. 3. A’s location information is not visible to C. 4. A cannot see the location of B or C. 5. After A revokes location sharing with B, B can no longer see the declarative text information. 6. A’s location information is not visible to B or C.
Deep inspection	1. When A changes location policy allowing B, a NOTIFY is sent from A to B over NNI that contains the new information. Any subsequent NOTIFY to C does not contain location information. 2. When A changes location policy revoking B, a NOTIFY is sent from A to B over NNI that in which the location information has been removed. Any NOTIFY to C does not contain location information. Note: The mechanism where A defines location privacy policy is at the UNI and is not visible at the NNI. Location information should not appear for any watcher over the NNI until the appropriate policy has been defined by the user at its XDMS.

Test case ID	**RCS_5_4_2**
Related Test Cases	**RCS_5_1_1, RCS_5_2_1, RCS_5_2_2, RCS_5_4_1**
Feature	Social Presence - Location management
Purpose	Social Presence: Coordinate information provided
Pre-conditions Scenario	1. Social Presence is enabled via Presence Server (PRESENCE PROFILE = 1). 2. A is sharing SPI with B, including location. (test **RCS_5_1_1**) 3. A and B are VIP contacts.
Test procedure	1. A defines A’s map location (e.g., adds a pin to a map). 2. A removes the map locations (removes / deletes pin from a map).
Expected results **Post-conditions**	1. A’s map location is displayed in B’s map display. 2. A’s map location is cleared from B’s EAB map display.
Deep inspection	1. When A defines location information, a NOTIFY is sent from A to B over NNI that contains the new information. 2. When A removes location information, a NOTIFY is sent from A to B over NNI from which the location information has been removed.

Test case ID	**RCS_5_4_3**
Related Test Cases	**RCS_5_1_1, RCS_5_2_1, RCS_5_2_2, RCS_5_4_1**
Feature	Social Presence - Location management
Purpose	Social Presence: Location – Time Zone
Pre-conditions Scenario	1. Social Presence is enabled via Presence Server (PRESENCE PROFILE = 1). 2. A is sharing SPI with B, including location. (test **RCS_5_1_1**) 3. Both operators support Time Zone, and devices under test support Time Zone, and the Time Zone has been defined for A and B. 4. A and B are VIP contacts.
Test procedure	A and B examine their SPI information for each other.
Expected results **Post-conditions**	A and B can see each other’s Time Zone.
Deep inspection	1. The NOTIFY messages between A and B will contain Time Zone information. 2. If either A or B changes his or her time zone, whether this is user or device driven, a NOTIFY will be generated which identifies the new Time Zone information.

5.5 *Multi-device handling*

Test case ID	**RCS_5_5_1**
Related Tests	**RCS_5_1_1**
Feature	Social Presence – Multi-device Handling
Purpose	Social Presence: Interaction – Multi-device Invitation
Pre-conditions Scenario	1. Social Presence is enabled via Presence Server (PRESENCE PROFILE = 1). 2. B has multiple devices (e.g., mobile and PC). 3. A is not sharing SPI with B.

Test procedure	1. A sends a SPI to B. 2. B receives the invitation on all of B's devices. 3. B accepts the invitation on one of devices.
Expected results **Post-conditions**	1. The invitation to share SPI is shown on all of B's devices. 2. B can accept the invitation on any device and then receive A's SPI on all of B's devices.
Deep inspection	1. A SIP SUBSCRIBE will be sent from A to B over NNI. 2. Response to SUBSCRIBE is 202 Accepted. This will be followed by a "dummy" NOTIFY with a pending subscription state. 3. After B accepts the invite: A. There will be a full NOTIFY from B to A with an active subscription state. This NOTIFY will contain B's SPI. B. User B's network will send a SUBSCRIBE over NNI which will result in 200 OK and NOTIFY for active subscription with A' SPI. There will be a SUBSCRIBE for each of B's active devices over the NNI for B's information.

6 File Transfer

6.1 File Transfer using MSRP

6.1.1 Basic File Transfer (MSRP)

Test case ID	**RCS_6_1_1_1**
Related test cases	
Feature	File Transfer - File Transfer (FT) using MSRP
Purpose	Successful file transfer (single file with single recipient)
Pre-conditions Scenario	1. A and B are FT capable. 2. A has B as an RCS contact. 3. B has enough free storage space to receive the file. 4. The file size being transferred is smaller than the warning file size.
Test procedure	1. A selects one file from the local storage and then selects the FT option. 2. A selects B from the list of active RCS contacts and selects to share. 3. B reviews and then accepts the invitation. 4. The file is transferred to B.

Expected results **Post-conditions**	1. A sees that FT is available for B. 2. B receives FT invitation, Including the size of the file and an indication of the file type (e.g., MIME icon or mini-preview). Upon B accepting the file transfer invitation, the file is transferred and received by B.
Deep inspection	1. Verify file transfer invitation process. 2. Verify file transfer session establishment. 3. Verify successful transfer completion and session termination.

Test case ID	**RCS_6_1_1_2**
Related test cases	**RCS_6_1_1_1**
Feature	File Transfer - FT using MSRP
Purpose	Successful multiple FTs (multiple files to single recipient)
Pre-conditions **Scenario**	1. A has B as an RCS contact. 2. B has enough free storage space to receive file(s). 3. The file size being transferred is smaller than the warning file size.
Test procedure	1. A selects three files from the local storage and then selects the FT option. 2. A selects B from the list of active RCS contacts and selects to share. 3. B reviews and then accepts the invitation. 4. The files are transferred to B.
Expected results **Post-conditions**	1. A sees that FT is available for B. 2. For each file, the following steps can be verified: a. B receives file transfer invitation. • B should see the size of the file and an indication of the file type (e.g., MIME icon or mini-preview). b. Upon B accepting the file transfer invitation, the file is transferred and received by B.
Deep inspection	1. Verify that the sessions are NOT established in parallel. 2. For each session: a. Verify file transfer invitation process. b. Verify file transfer session establishment. c. Verify successful transfer completion and session termination.

Test case ID	**RCS_6_1_1_3**
Related test cases	
Feature	File Transfer - FT using MSRP
Purpose	Successful FT (single file with multiple recipients using conference focus)
Pre-conditions Scenario	1. A has B, C and D as RCS contacts. 2. All users have enough free storage space to receive file. 3. The file size being transferred is smaller than the warning file size.
Test procedure	1. A selects one file from the local storage and then selects the FT option. 2. A selects B, C and D from the list of active RCS contacts and selects to share. 3. B, C, and D review and then accept the invitation. 4. The file is transferred to B, C, and D.
Expected results **Post-conditions**	For B, C, and D, the following steps can be verified: 1. A sees that file transfer is available for each recipient. 2. The recipient receives file transfer invitation. 3. The recipient should see the size of the file and an indication of the file type (e.g., MIME icon or mini-preview). 4. Upon the recipient accepting the file transfer invitation, the file is transferred and received. 5. The recipient can access the file (e.g., from storage on the handset).
Deep inspection	1. Verify that only one session is established from the sender to conference focus and one terminating session to each recipient from the focus. 2. For each terminating session: a. Verify file transfer invitation process. b. Verify file transfer session establishment. c. Verify successful transfer completion and session termination.

Test case ID	**RCS_6_1_1_4**
Related test cases	
Feature	File Transfer - File Transfer using MSRP
Purpose	Successful multiple FTs (multiple files to multiple recipients)
Pre-conditions Scenario	1. A has B and C as RCS contacts. 2. All users have enough free storage space to receive file(s). 3. The file size being transferred is smaller than the warning file size. 4. A, B, and C are in an active group chat session.

Test procedure	1. A selects three files from the local storage and then selects the file transfer option. 2. A selects B and C from the list of contacts and selects to share. 3. B and C review and then accept the invitation. 4. The files are transferred to B and C.
Expected results **Post-conditions**	For each user, the following steps can be verified: 1. A sees that file transfer is available for each recipient. 2. For each file, the following steps can be verified: a. The recipient receives file transfer invitation. b. The recipient should see the size of the file and an indication of the file type (e.g., MIME icon or mini-preview). c. Upon the recipient accepting the file transfer invitation, the file is transferred and received.
Deep inspection	1. Verify that only one session is established from the sender and one terminating session to each recipient. 2. For each terminating session: a. Verify file transfer invitation process. b. Verify file transfer session establishment. c. Verify successful transfer completion and session termination.

Test case ID	**RCS_6_1_1_5**
Related test cases	
Feature	File Transfer - File Transfer using MSRP
Purpose	Successful simultaneous sessions, one for FT and second one for chat.
Pre-conditions Scenario	1. A has B and C as RCS contacts. 2. All users have enough free storage space to receive file(s). 3. The file size being transferred is smaller than the warning file size. 4. A, B and C are in an active group chat session.
Test procedure	1. A selects the FT option. 2. A selects one file, and selects the share option. 3. B accepts the invitation and opens the file. 4. B sends an IM/Chat invitation to A. 5. A accepts the IM/Chat invitation. 6. IM/Chat session starts.
Expected results **Post-conditions**	1. B receives file. 2. B can open the file properly. 3. A receives the IM/Chat invitation. 4. The IM/Chat session is established. For A->B leg verify: 1. B receives the invitation stating the file type and the size. 2. If it is a supported file type, B should see the relevant MIME icon or mini-preview within the chat window (all FTs take place in a chat window).

	3. A can progress the transfer in the notification bar (the transfer does not block the UI). 4. B receives the file. 5. B can open the file properly.
Deep inspection	For file transfer session: 1. Verify file transfer invitation process; FT session is separate from chat sessions 2. Verify file transfer session establishment. 3. Verify successful transfer completion and session termination.

Test case ID	**RCS_6_1_1_6**
Related test cases	
Feature	File Transfer - File Transfer using MSRP
Purpose	Verifies correct operation when FT is rejected.
Pre-conditions Scenario	1. A has B as an RCS contact. 2. The file size being transferred is smaller than the warning file size.
Test procedure	1. A selects one file from the local storage and then selects the file transfer option. 2. A selects B from the list of active RCS contacts and selects to share. 3. B rejects the invitation.
Expected results **Post-conditions**	1. B receives the file transfer invitation. 2. The file is not received by B. 3. A is notified that B has rejected the file transfer request.
Deep inspection	1. Verify file transfer invitation process. 2. Verify file transfer session is not established with an error code 603 Decline response.

Test case ID	**RCS_6_1_1_7**
Related test cases	
Feature	File Transfer - File Transfer using MSRP

Purpose	Successful auto-accept of the FT session (fast acceptance)
Pre-conditions Scenario	1. Capabilities exchange has taken place confirming FT share is possible. 2. The ftAutAccept parameter on B's client is set to 1.
Test procedure	1. A selects B. 2. A selects file transfer option. 3. A selects one file and selects share option.
Expected results **Post-conditions**	1. B receives file. 2. B can open the file correctly.
Deep inspection	1. Verify capability exchange. 2. Verify file transfer invitation auto-accept. 3. Verify file transfer session establishment. 4. Verify successful transfer completion and session termination.

Test case ID	**RCS_6_1_1_8**
Related test cases	
Feature	File Transfer - File Transfer using MSRP
Purpose	Ongoing FT cancelled by the sender.
Pre-conditions Scenario	All users have enough free storage space to receive file.
Test procedure	1. A selects B. 2. A selects the FT option. 3. A selects a file. 4. B accepts the invitation. 5. A cancels file transfer while the transferring is going on.

Expected results **Post-conditions**	1. B receives the file transfer invitation. 2. A cancelled the file transfer; B receives an error message.
Deep inspection	1. Verify file transfer invitation process. 2. Verify file transfer session establishment. 3. Verify cancelation and session termination with a SIP BYE from A.

Test case ID	**RCS_6_1_1_9**
Related test cases	**RCS_6_1_1_8**
Feature	File Transfer - File Transfer using MSRP
Purpose	Ongoing FT cancelled by the recipient.
Pre-conditions **Scenario**	All users have enough free storage space to receive file(s).
Test procedure	1. A selects B from his/her address book. 2. A selects the FT option. 3. A selects three files. 4. B accepts the invitation. 5. B cancels the file transfer while the third file is being transferred.
Expected results **Post-conditions**	1. B receives the file transfer invitation. 2. B receives first and second files. 3. B cancelled the third file transfer and A received an error message. 4. B can open received first two files properly.
Deep inspection	1. Verify capability exchange. 2. Verify file transfer invitation process. 3. Verify file transfer session establishment. 4. Verify cancellation and session termination with a SIP BYE from B before the third file is completely transferred.

Test case ID	**RCS_6_1_1_10**
Related test cases	

Feature	File Transfer - File Transfer using MSRP
Purpose	Unsuccessful transfer: Recipient does not answer.
Pre-conditions Scenario	1. B has auto-accept disabled. 2. A and B have enough free storage space to receive file(s).
Test procedure	1. A selects B. 2. A selects the FT option. 3. A selects a file. 4. B does NOT answer the file transfer invitation.
Expected results **Post-conditions**	1. B receives the file invitation. 2. A receives an error message after timeout.
Deep inspection	1. Verify capability exchange. 2. Verify the file transfer invitation process. 3. Verify SIP INVITE timeout.

Test Case ID	**RCS_6_1_1_11**
Related Test Cases	**RCS_6_1_1_12**
Feature	File Transfer - File Transfer using MSRP
Purpose	Interrupted FT resumed by recipient
Pre-conditions Scenario	1. A and B are capable of RCS MSRP-based file transfer. 2. A and B are shown as available. 3. A and B use manual acceptance of file transfer.
Test procedure	1. A sends a file to B. (Note: A sends a large file to allow interrupted during file transfer.) 2. B accepts the invitation and the file transfer starts. 3. The file transfer is interrupted while it is being sent (e.g., A or B loses connection momentarily or network problem). 4. B resumes the file transfer.

Expected results **Post-conditions**	1. When B resumes the file transfer (that had been interrupted), A receives a file transfer invitation from B. Note: The resumption strategy is Service Provider policies - such as how quickly the resume will be initiated, how many retries, etc., and it is based on error code. 2. If A supports the file transfer resume operation, a. A accepts the invitation and the file transfer is resumed from where it was interrupted. b. B receives the entire file. 3. If A does not support file transfer resume operation, a. A rejects the invitation. b. B is informed of the file transfer failure.
Deep inspection	1. Verify that A receives a SIP INVITE from B for resumption, which includes the file-range attribute, including the file-selector for the missing part. 2. Verify that if A does not support resume operation, the SIP INVITE for the resumption from B is rejected.

Test Case ID	**RCS_6_1_1_12**
Related Test Cases	**RCS_6_1_1_11**
Feature	File Transfer - File Transfer using MSRP
Purpose	Interrupted file transfer resumed by sender
Pre-conditions **Scenario**	1. A and B are capable of RCS MSRP-based file transfer. 2. A and B are shown as available. 3. A and B use manual acceptance of file transfer.
Test procedure	1. A sends a file to B. (Note: A send a large file to allow interruption during file transfer.) 2. B accepts the invitation and file transfer starts. 3. During the file transfer is interrupted while it is being sent (e.g., A or B loses connection momentarily or network problem). 4. A resumes the file transfer.
Expected results **Post-conditions**	1. When A resumes the file transfer (that had been interrupted), B receives a file transfer invitation from A. Note: The resumption strategy is Service Provider policies- such as how quickly the resumption will be initiated, how many retries, etc, and it is based on error code. 2. If B client supports the file transfer resume operation, a. B accepts the invitation and returns 200 OK with SDP including the correct file-range attribute with file selector attribute. b. The file transfer is resumed from where it was interrupted. c. B receives the entire file successfully. 3. If B client does not support file transfer resume operation, a. B client rejects the invitation.

	b. A may be informed of the file transfer failure (UI implementation).
Deep inspection	1. Verify that B receives the SIP INVITE from A for resumption, which includes the best estimated file-range attribute and the file-selector attribute for the missing part. 2. Verify that B responds 200 OK to the resumption INVITE with a correct file-range attribute and file selector attribute. 3. Verify that if B does not support resumption, the SIP INVITE for the resumption from A is rejected.

6.1.2 Multi-device handling

Test Case ID	**RCS_6_1_2_1**
Related Test Cases	**RCS_6_1_1_11, RCS_6_1_1_12**
Feature	File Transfer - File Transfer using MSRP
Purpose	File transfer resumption request sent to the device that originates the file transfer (Multi-device Handling).
Pre-conditions Scenario	1. A and B are capable of RCS MSRP-based file transfer. 2. A and B are shown as available. 3. A and B use manual acceptance of file transfer. 4. A and B clients supports the file transfer resumption. 5. A has two devices: A1 and A2. Both A1 and A2 are online.
Test procedure	1. A1 sends a file to B. (Note: A1 send a large file to allow interruption during file transfer.) 2. B accepts the invitation and the file transfer starts. 3. The file transfer is interrupted while it is being sent (e.g., A1 or B loses connection momentarily or network problem). 4. B resumes the file.
Expected results **Post-conditions**	1. When B resumes the file transfer (that had been interrupted), A1 receives a file transfer invitation from B. 2. When A1 accepts the invitation, A1 resumes the file transfer. 3. B receives the file successfully.
Deep inspection	1. Verify that the device identifier (sip.instance or GRUU) is included in the SIP INVITE for the original file transfer from A1 to B. 2. Verify that the same device identifier is used to address the sender in the SIP INVITE of the resumption request from B. 3. Verify that A1 receives the SIP INVITE from B for resumption. 4. If the device identifier is not included in the original file transfer SIP INVITE from A1 to B, no device identifier will be included in the SIP INVITE of the resumption request from B. The resumption SIP INVITE will be forked to A1

	and A2. A1 will answer the SIP INVITE with 200 OK.

6.1.3 Exception conditions

Test case ID	**RCS_6_1_3_1**
Related test cases	
Feature	File Transfer - File Transfer using MSRP
Purpose	Unsuccessful transfer before completion: Sender loses coverage.
Pre-conditions Scenario	All users have enough free storage space to receive file(s).
Test procedure	1. A selects B. 2. A selects the FT option. 3. A selects three files and selects the share option. 4. B accepts the invitation. 5. A loses coverage while third file is being transferred.
Expected results **Post-conditions**	1. B receives the files invitation. 2. B receives first and second files. 3. B receives an error message that A has interrupted file transfer. 4. B receives updated capabilities for A. 5. B can open first two received files properly.
Deep inspection	1. Verify capability exchange. 2. Verify file transfer invitation process. 3. Verify file transfer session establishment. 4. Verify session timeout.

Test case ID	**RCS_6_1_3_2**
Related test cases	
Feature	File Transfer - File Transfer using MSRP
Purpose	Unsuccessful transfer before completion: Recipient loses coverage.

Pre-conditions Scenario	All users have enough free storage space to receive file(s).
Test procedure	1. A selects B. 2. A selects the file transfer option. 3. A selects three files and selects the share option. 4. B accepts the invitation. 5. B loses coverage while the third file is being transferred.
Expected results **Post-conditions**	1. B receives files invitation. 2. B receives first two files. 3. A receives an error message that B has interrupted file transference. 4. A receives updated capabilities for B. 5. B can open the two received files properly.
Deep inspection	1. Verify capability exchange. 2. Verify file transfer invitation process. 3. Verify file transfer session establishment. 4. Verify session timeout.

Test case ID	**RCS_6_1_3_3**
Related test cases	
Feature	File Transfer - File Transfer using MSRP
Purpose	File size warning limit (terminating side) (for NNI testing, ensure File Transfer file size attribute is carried over the NNI and available to B to invoke FT warning procedures).
Pre-conditions Scenario	1. Capabilities exchange between A and B has taken place confirming file transfer is possible. 2. FT AUT ACCEPT on B is set to 1. 3. FT WARN SIZE in KB on B is set to a value below the size of the file to be sent. 4. FT MAX SIZE in KB on both A and B is set to a value above the size of the file to be sent.
Test procedure	A selects a file to share with B that is larger than the FT WARN SIZE for B, but smaller than the FT MAX SIZE on both A and B.

Expected results **Post-conditions**	1. A gets a warning message if the FT WARN SIZE parameter is also configured for A and confirmation is asked to proceed. 2. A confirms and proceeds. 3. Instead of B's device auto-accepting the file, B is receives the file size and B is asked to accept the File Transfer invitation. 4. After about 5 seconds, B accepts the File Transfer invitation.
Deep inspection	1. Verify file transfer invitation process. 2. Verify that B is asked to accept the invitation because the file size is greater than the FT WARN SIZE and no auto-accept is performed.

Test case ID	**RCS_6_1_3_4**
Related test cases	
Feature	File Transfer - File Transfer using MSRP
Purpose	Maximum File size exceeded (terminating side) (for NNI testing, ensure File Transfer file size attribute is carried over NNI and available to B's network to invoke FT maximum size exceeded procedures).
Pre-conditions **Scenario**	1. Capabilities exchange between A and B has taken place confirming file transfer is possible. 2. FT MAX SIZE in KB on A is greater than FT MAX SIZE in KB on B.
Test procedure	A selects a file to send to B that is smaller than the FT MAX SIZE on A but larger than the FT MAX SIZE on B.
Expected results **Post-conditions**	B detects that the file to be transferred is too big (i.e., a=file-selector size value is larger than FT MAX SIZE on B), so B answers the INVITE with a 488 “Not Acceptable Here” and warning header set to “133 Size exceeded,” OR Network B's Messaging Server detects that the file exceeds the maximum file transfer size and returns an error a 403 “Forbidden” and warning header set to “133 Size exceeded.”
Deep inspection	1. Verify the a=file-selector size attribute in the incoming SIP request. 2. If B detects that the file to be transferred is too big (i.e., a=file-selector size value is larger than FT MAX SIZE on B), verify that B answers the INVITE with a 488 “Not Acceptable Here” and warning header set to “133 Size exceeded.” 3. If Network B's Messaging Server detects that the file exceeds the maximum file transfer size verify that it returns an error a 403 “Forbidden” and warning header set to “133 Size exceeded.”

6.2 File transfer using HTTP

6.2.1 Basic File Transfer (HTTP)

For the following test cases, an HTTP proxy on the recipient side and a dedicated link (e.g., VPN) between operators are assumed, so that HTTP traffic will also be considered as NNI traffic.

Note: If this is not the case, the HTTP traffic is not seen on the NNI and therefore is not part of the deep inspection of the test case.

Test Case ID	**RCS_6_2_1_1**
Related test cases	
Feature	File Transfer - File Transfer using HTTP
Purpose	Successful HTTP-based file transfer in active one-to-one chat.
Pre-conditions Scenario	1. A and B are in an active one-to-one chat session. 2. B has enough free storage space to receive file. 3. File size being transferred is smaller than the warning file size. 4. A and B support HTTP-based FT according to former capability exchange. 5. A and B parameters: a. FT MAX SIZE is the same for A and B. b. Preferred File Transfer technology is HTTP.
Test procedure	1. A selects one file from the local storage and then selects the file transfer option. 2. B receives and accepts the file download notification if prompted. 3. B opens the file.
Expected results **Post-conditions**	1. A sees that file transfer option is available for B. 2. B receives file transfer download notification and sees the size of the file and an indication of the file type (e.g., MIME icon or mini-preview). 3. The file is transferred and received. B can access the file (e.g., from storage on the handset). 4. Display notification is received and shown on A.
Deep inspection	1. Verify that capability exchange, including HTTP FT tag, occurs before chat session setup. 2. Verify support of FT HTTP XML content type during SDP negotiation. 3. Verify that FT notification XML is sent inside the active chat. 4. Verify that the URL complies to ftcontentserver.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org. 5. Verify that HTTPS requests for file and optional thumbnail are sent to the content server in A's network and the content is received. 6. Verify display notification in SIP MESSAGE. 7. Note: It is assumed that the file will always upload successfully.

Test Case ID	**RCS_6_2_1_2**

Related test cases	**RCS_6_2_1_1**
Feature	File Transfer - File Transfer using HTTP
Purpose	Successful HTTP-based file transfer in new one-to-one chat.
Pre-conditions Scenario	1. A has B as RCS contact. 2. A and B are not in an active one-to-one chat session. 3. B has enough free storage space to receive file. 4. File size being transferred is smaller than the warning file size. 5. A and B support HTTP-based FT. 6. A parameters: a. FT MAX SIZE is the same for A and B. b. Preferred File Transfer technology is HTTP. 7. B parameters: a. FT MAX SIZE is same for A and B. b. Preferred File Transfer technology is HTTP. c. Standalone message enabled is set to false.
Test procedure	1. A selects one file from the local storage and then selects the file transfer option. 2. A selects B and selects to share. 3. B receives and accepts the chat invitation and file download notification if prompted. 4. B opens the file.
Expected results **Post-conditions**	1. A sees that file transfer option is available for B. 2. B receives file transfer download notification and sees the size of the file and an indication of the file type (e.g. MIME icon or mini-preview). 3. The file is transferred and received. B can access the file (e.g., from storage on the handset). 4. Display notification is received and shown on A.
Deep inspection	1. Verify that capability exchange including HTTP FT tag, but without standalone message capability occurs before chat session setup. 2. Verify that a new one-to-one chat session is created carrying the HTTP FT tag in the accept contact header and used to send the FT notification to B. 3. Verify support of FT HTTP XML content type during SDP negotiation. 4. Verify that the URL complies to ftcontentserver.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org. 5. Verify that HTTPS requests for file and optional thumbnail are sent to the content server in A's network and the content is received. Note: It is assumed that the file will always upload successfully.

Test Case ID	**RCS_6_2_1_3**
Related test cases	**RCS_6_2_1_1**
Feature	File Transfer - File Transfer using HTTP
Purpose	Successful HTTP-based file transfer in one-to-one standalone message

Pre-conditions Scenario	1. A has B as RCS contact. 2. A and B are not in an active 1-2-1 chat session 3. B has enough free storage space to receive file. 4. File size being transferred is smaller than the warning file size. 5. A and B support HTTP-based FT and standalone messaging. 6. A and B parameters: a. FT MAX SIZE is the same for A and B. b. Preferred File Transfer technology is HTTP. c. Standalone message enabled is set to true.
Test procedure	1. A selects one file from the local storage and then selects the file transfer option. 2. A selects B and selects to share. 3. B receives and accepts the file download notification if prompted. 4. B opens the file.
Expected results **Post-conditions**	1. A sees that file transfer option is available for B. 2. B receives file transfer download notification and sees the size of the file and an indication of the file type (e.g., MIME icon or mini-preview). 3. The file is transferred and received. B can access the file (e.g., from storage on the handset). 4. Display notification is received and shown on A.
Deep inspection	1. Verify that capability exchange, including HTTP FT and standalone message tags, occurs before FT. 2. Verify that the FT notification XML is sent in a standalone message. 3. Verify that the URL complies to ftcontentserver.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org. 4. Verify that HTTPS requests for file and optional thumbnail are sent to the content server on A's network and content is received. 5. Verify display notification in SIP MESSAGE. Note: It is assumed that the file will always upload successfully.

Test Case ID	**RCS_6_2_1_4**
Related test cases	
Feature	File Transfer - File Transfer using HTTP
Purpose	Successful one-to-many HTTP-based file transfer in active group chat (GC focus on A side)
Pre-conditions Scenario	1. A, B, and C are in an active group chat session. 2. Group chat is hosted by MNO A side. 3. B and C have enough free storage space to receive file. 4. File size being transferred is smaller than the warning file size. 5. Clients A, B, and C support HTTP-based FT. 6. FT MAX SIZE is the same for A, B and C. 7. A parameters: a. Preferred File Transfer technology is HTTP. b. File transfer in group chat is supported. 8. B and C parameters: a. Preferred File Transfer technology is HTTP.

Test procedure	1. A selects one file from the local storage and starts file transfer to the group chat. 2. B and C receive and accept the file download notification. 3. B and C open the file.
Expected results **Post-conditions**	1. A sees that file transfer is available. 2. B and C receive the file download notification and each sees the size of the file and an indication of the file type (e.g., MIME icon or mini-preview). 3. The file is transferred and received by B and C. The recipients can access the file (e.g., from storage on the handset). 4. Display notification is received and shown on client A for delivery to B and C.
Deep inspection	1. Verify capability exchange during session setup, including HTTP FT tag, in contact header within SIP INVITE and 200 OK between GC focus A and B, C. 2. Verify support of FT HTTP XML content type during SDP negotiation. 3. Verify that the FT notification XML is sent inside the active chat from A to B and C. 4. Verify that the URL complies to ftcontentserver.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org. 5. Verify that HTTPS requests for file and optional thumbnail are sent by B and C to the content server in A's network and the content is received. Note: It is assumed that the file will always upload successfully.

Test Case ID	**RCS_6_2_1_5**
Related test cases	**RCS_6_2_1_4**
Feature	File Transfer - File Transfer using HTTP
Purpose	Successful HTTP-based file transfer in active group chat (GC focus on B side)
Pre-conditions **Scenario**	1. A, B, and C are in an active group chat session. 2. Group chat is hosted by MNO B side. 3. B and C have enough free storage space to receive file. 4. File size being transferred is smaller than the warning file size. 5. Clients A, B, and C support HTTP-based FT. 6. A and C parameters: a. FT MAX SIZE is the same as for A, B and C. b. Preferred File Transfer technology is HTTP. 7. B parameters: a. FT MAX SIZE is the same for A, B and C. b. Preferred File Transfer technology is HTTP. c. File transfer in group chat is supported.
Test procedure	1. A selects one file from the local storage and starts file transfer to the group chat. 2. B and C receive and accept the file download notification if prompted. 3. B and C open the file.

Expected results **Post-conditions**	1. A sees that file transfer option is available for B and C. 2. B and C receive the file download notification and each sees the size of the file and an indication of the file type (e.g., MIME icon or mini-preview). 3. The file is transferred and received by B and C. The recipients can access the file (e.g., from storage on the handset). 4. Display notification is received and shown on A for delivery to B and C.
Deep inspection	1. Verify capability exchange during session setup including HTTP FT tag in contact header within SIP INVITE and 200 OK between A, C and GC focus server B. 2. Verify support of FT HTTP XML content type during SDP negotiation. 3. Verify that the FT notification XML is sent inside the active chat from to B and C. 4. Verify that the URL complies to ftcontentserver.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org. 5. Verify that HTTPS requests for file and optional thumbnail are sent by B and C to the content server A and the content is received. Note: It is assumed that the file will always upload successfully.

Test Case ID	**RCS_6_2_1_6**
Related test cases	**RCS_6_2_1_4**
Feature	File Transfer - File Transfer using HTTP
Purpose	Successful HTTP-based file transfer in new group chat (GC focus on A side)
Pre-conditions Scenario	1. A has B and C as RCS contacts. 2. A, B and C are in not in an active group chat session. 3. B and C have enough free storage space to receive file. 4. File size being transferred is smaller than the warning file size. 5. A, B, and C support HTTP-based FT. 6. A parameters: a. FT MAX SIZE is the same for A, B and C. b. Preferred File Transfer technology is HTTP. c. Standalone message enabled is set to false. d. File transfer in group chat is supported. 7. B and C parameters: a. FT MAX SIZE is the same for A, B and C. b. Preferred File Transfer technology is HTTP. c. Standalone message enabled is set to false (for B or C).
Test procedure	1. A selects one file from the local storage and then selects the file transfer option. 2. A selects B and C and selects to share. 3. B and C receive and accept the group chat invitation and file download notification if prompted. 4. B and C open the file.

Expected results **Post-conditions**	1. A sees that the file transfer option is available for B and C. 2. B and C receive file download notification and each sees the size of the file and an indication of the file type (e.g., MIME icon or mini-preview). 3. The file is transferred and received by B and C. The recipients can access the file (e.g., from storage on the handset). 4. Display notification is received and shown on A for delivery to B and C.
Deep inspection	1. Verify that capability exchange including FT HTTP tag but without standalone message tag for A, B, or C occurs before file transfer. 2. Verify capability exchange during session setup including HTTP FT tag in contact header within SIP INVITE and 200 OK between GC focus and clients B and C. 3. Verify support of FT HTTP XML content type during SDP negotiation. 4. Verify that a new chat is created and used to send the file transfer notification to clients B and C. 5. Verify that the URL complies to ftcontentserver.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org. 6. Verify that chat session is closed afterwards. 7. Verify that HTTPS requests for file and optional thumbnail are sent by B and C to the content server in A's network and the content is received. Note: It is assumed that the file will always upload successfully.

Test Case ID	**RCS_6_2_1_7**
Related test cases	
Feature	File Transfer - File Transfer using HTTP
Purpose	Successful HTTP-based file transfer in one-to-many standalone message
Pre-conditions **Scenario**	1. A has users B, C as RCS contacts. 2. A, B and C are in not in an active group chat session. 3. B and C have enough free storage space to receive file. 4. File size being transferred is smaller than the warning file size. 5. A, B and C support HTTP-based FT and standalone messaging. 6. A, B and C parameters: a. FT MAX SIZE is the same for A, B and C. b. Preferred File Transfer technology is HTTP. c. File transfer in group chat is supported. d. Standalone message enabled is set to true.
Test procedure	1. A selects one file from the local storage and then selects the file transfer option. 2. A selects B and C and selects to share. 3. B and C receive and accept the file download notification if prompted. 4. B and C open the file.

Expected results **Post-conditions**	1. A sees that file transfer option is available for the recipients B and C. 2. B and C receive file download notification and each sees the size of the file and an indication of the file type (e.g., MIME icon or mini-preview). 3. The file is transferred and received by B and C. The recipients can access the file (e.g., from storage on the handset). 4. Display notification is received and shown on A for delivery to B and C.
Deep inspection	1. Verify that capability exchange including HTTP FT and standalone message tags between A, B and C occurs. 2. Verify that the FT notification is sent in a standalone message to B and C. 3. Verify that the URL complies to ftcontentserver.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org. 4. Verify that HTTPS requests for file and optional thumbnail are sent by B and C to the content server in A's network and content is received. Note: It is assumed that the file will always upload successfully.

Test Case ID	**RCS_6_2_1_8**
Related test cases	
Feature	File Transfer - File Transfer using HTTP
Purpose	Successful HTTP-based file transfer in active one-to-one chat (resume file download)
Pre-conditions Scenario	1. A and B are in active one-to-one chat session. 2. B has enough free storage space to receive file. 3. File size being transferred is smaller than the warning file size. 4. A and B support HTTP-based FT. 5. A parameters: a. FT MAX SIZE is same for A and B. b. Preferred File Transfer technology is HTTP. c. File Transfer Resume supported is set to true. 6. B parameters: a. FT MAX SIZE is the same for A and B. b. Preferred File Transfer technology is HTTP.
Test procedure	1. A selects one file from the local storage and then selects the file transfer option to B. 2. B receives and accepts the file download notification if prompted. 3. The file transfer to B is started. 4. During file download, B gets out of coverage. 5. B resumes download after returning to coverage. 6. B opens the file.
Expected results **Post-conditions**	1. A sees that file transfer is available for the recipient. 2. B receives file transfer notification and sees the size of the file and an indication of the file type (e.g., MIME icon or mini-preview). 3. The download of the file starts but stops before the transfer is complete. 4. When B is back in data coverage, the transfer proceeds until the file received by B. 5. Display notification is received and shown on A (sent by B).

Deep inspection	1. Verify that capability exchange including HTTP FT tag occurs before chat session. 2. Verify support of FT HTTP XML content type during SDP negotiation. 3. Verify that FT notification XML is sent inside the active chat. 4. Verify that the URL complies to ftcontentserver.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org. 5. Verify that HTTPS requests for file and optional thumbnail are sent to the content server in A's network and first content is received. 6. Verify that HTTPS requests for file are resumed (and not restarted) when client gets back into coverage. 7. Verify display notification in SIP MESSAGE. Note: It is assumed that the file will always upload successfully.

6.2.2 Multi-device handling

Test Case ID	**RCS_6_2_2_1**
Related test cases	**RCS_6_2_1_1**
Feature	File Transfer - File Transfer using HTTP
Purpose	Successful HTTP-based file transfer in active one-to-one chat with multiple clients on A side.
Pre-conditions Scenario	1. A has two registered and online clients A1 and A2. 2. A1 and B are in active one-to-one chat session. 3. B has enough free storage space to receive file. 4. File size being transferred is smaller than the warning file size. 5. Clients A1, A2 and B support HTTP-based FT according to former capability exchange. 6. Network A and B NNI parameters: a. FT MAX SIZE is the same for A and B. b. Preferred File Transfer technology is HTTP.
Test procedure	1. A1 selects one file from the local storage and then selects the file transfer option. 2. B receives and accepts the file download notification if prompted. 3. B opens the file.
Expected results **Post-conditions**	1. A1 sees that file transfer option is available for B. 2. B receives file transfer notification and sees the size of the file and an indication of the file type (e.g., MIME icon or mini-preview). 3. The file is transferred and received by B. 4. Display notification is received and shown only on client A1 (but not on client A2).
Deep inspection	1. Verify that capability exchange, including HTTP FT tag, occurs before chat session. 2. Verify support of FT HTTP XML content type during SDP negotiation. 3. Verify that FT notification XML is sent inside the active chat. 4. Verify that the URL complies to

	ftcontentserver.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org. 5. Verify that HTTPS requests for file and optional thumbnail are sent to the content server on A's network and the content is received. 6. Verify display notification in SIP MESSAGE. The accept-contact header shall address the device ID (sip.instance or GRUU) of client A1 out of the contact header during session setup. Note: It is assumed that the file will always upload successfully.

6.2.3 Exception conditions

Test Case ID	**RCS_6_2_3_1**
Related test cases	
Feature	File Transfer - File Transfer using HTTP
Purpose	Fallback to MSRP-based file transfer (one-to-many) due to missing GC focus support
Pre-conditions Scenario	1. A, B, and C are in an active group chat session. 2. Group chat is hosted by MNO B side and the focus does not support HTTP based FT. 3. B and C have enough free storage space to receive file. 4. File size being transferred is smaller than the warning file size. 5. Clients A, B, and C support HTTP-based FT. 6. A and C parameters: a. FT MAX SIZE is the same as for A, B and C. b. Preferred File Transfer technology is HTTP. c. File Transfer Thumbnail is supported. 7. Network B NNI parameters: a. FT MAX SIZE is the same for A, B and C. b. Preferred File Transfer technology is MSRP. c. File Transfer Thumbnail is supported. d. File transfer in group chat is not supported.
Test procedure	1. A selects one file from the local storage and starts the file transfer to the group chat. 2. B and C receive and accept the file transfer invitation if prompted. 3. B and C open the file.
Expected results **Post-conditions**	1. A sees that file transfer is available for the group chat. 2. B and C receive the file transfer invitation and each sees the size of the file and an indication of the file type (e.g. MIME icon or mini-preview). 3. The file is transferred and received by B and C sequentially. The recipients can access the file (e.g., from storage on the handset). 4. Display notification is received and shown on A for delivery to B and C.
Deep inspection	1. Verify that capability exchange including FT and FT thumb tags, occurs between A, B, and C before chat session. 2. Verify capability exchange occurs during session setup without HTTP FT tag in contact header within SIP INVITE and 200 OK between GC focus server B and A.

	3. Verify that two additional MSRP sessions are created and used to send the file to B and C. 4. Verify that the two MSRP sessions are closed afterwards. Note: It is assumed that the file will always upload successfully.

Test Case ID	**RCS_6_2_3_2**
Related test cases	
Feature	File Transfer; File Transfer using HTTP
Purpose	Partially successful HTTP-based file transfer in active group chat (one client does not support HTTP-based FT).
Pre-conditions Scenario	1. A, B, and C are in an active group chat session. 2. Group chat is hosted by MNO A side. 3. B and C have enough free storage space to receive a file. 4. File size being transferred is smaller than the warning file size. 5. A and B support HTTP-based FT and C supports MSRP-based FT only. 6. A and B parameters: a. FT MAX SIZE is the same for A, B and C. b. Preferred File Transfer technology is HTTP. c. File transfer in group chat is supported. 7. C parameters: a. FT MAX SIZE same as for A. b. Preferred File Transfer technology is MSRP.
Test procedure	1. A selects one file from the local storage and starts the file transfer to the group chat. 2. B receives and accepts the file download notification if prompted. 3. B opens the file.
Expected results **Post-conditions**	1. A sees that the file transfer option is available for the recipients B and C. 2. B receives file transfer download notification and sees the size of the file and an indication of the file type (e.g., MIME icon or mini-preview). 3. C does not receive the file transfer invitation. 4. The file is transferred and received by B. The recipient can access the file (e.g., from storage on the handset). 5. Display notification is received and shown on A for delivery to B only.
Deep inspection	1. Verify capability exchange during session setup in contact header within SIP INVITE and 200 OK between GC focus A, B and C. GC focus A and A and B shall include HTTP FT tag. Client C shall include FT, FT thumb tags. 2. Verify support of FT HTTP XML content type during SDP negotiation by GC focus and A and B. 3. Verify that the FT notification XML is sent inside the active chat to B only. 4. Verify that the URL complies to ftcontentserver.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org 5. Verify that HTTPS requests for file and optional thumbnail are sent by B to content server on A's network and content is received. 6. Verify display notification in SIP MESSAGE is received from B.

	Note: It is assumed that the file will always upload successfully.

Test Case ID	**RCS_6_2_3_3**
Related test cases	**RCS_6_2_1_2**
Feature	File Transfer - File Transfer using HTTP
Purpose	Unsuccessful HTTP-based file transfer in new one-to-one chat (validity expired).
Pre-conditions Scenario	1. A has B as RCS contact. 2. A and B are not in active 1-2-1 chat session. 3. B has enough free storage space to receive file. 4. File size being transferred is smaller than the warning file size. 5. A and B support HTTP-based FT. 6. Stand-alone messaging is not supported by A or B. 7. A parameters: a. FT MAX SIZE is the same for A and B. b. Preferred File Transfer technology is HTTP. c. File Transfer to offline users allowed it set to true. d. Chat to offline users allowed is set to true. e. Standalone message enabled is set to false. f. File Transfer HTTP retention period less than IM S&F period on B side. 8. B parameters: a. FT MAX SIZE is the same for A and B. b. Preferred File Transfer technology is HTTP. c. File Transfer to offline users allowed it set to true. d. Chat to offline users allowed is set to true. e. Standalone message enabled is set to false.
Test procedure	1. A selects one file from the local storage and then selects the file transfer option. 2. A selects B and selects to share. 3. B is switched online after FT validity period (NNI File Transfer HTTP retention period) but before chat Store and Forward period. 4. B receives the (stored) file download notification and sees an error (file expired).
Expected results **Post-conditions**	1. RCS capability exchange takes place and A sees that file transfer option is available for the recipient. 2. After becoming online, B receives file transfer download notification stored on MNO B's Message Server and sees the size of the file and an indication of the file type (e.g., MIME icon but no mini-preview). 3. The file is not transferred and received. 4. No display notification is and shown on A.

Deep inspection	1. Verify that capability exchange, including HTTP FT tag but without standalone message capability, occurs before chat session. 2. Verify support of FT HTTP xml content type during SDP negotiation. 3. Verify that a new one-to-one chat session is created and used to send the FT notification to B. 4. In case of support for first message in INVITE, verify that FT notification is sent in INVITE. 5. Verify that the URL complies to ftcontentserver.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org. 6. Verify that chat session is closed afterwards. 7. Verify that the file is invalid due to timeout. 8. Depending on client implementation, verify (a) that no HTTPS requests for file and optional thumbnail are sent to the content server in A's network or (b) an HTTP error is sent back from the content server in A's network. 9. Verify display notification in SIP MESSAGE. Note: It is assumed that the file will always upload successfully.

Test Case ID	**RCS_6_2_3_4**
Related test cases	**RCS_6_2_1_2**
Feature	File Transfer - File Transfer using HTTP
Purpose	Unsuccessful HTTP-based file transfer in active one-to-one chat (file size exceeded).
Pre-conditions Scenario	1. A and B are in an active one-to-one chat session. 2. B has enough free storage space to receive file. 3. The file size being transferred is larger than FT MAX SIZE of MNO B but smaller than the warning file size (FT WARN SIZE) of MNO A. 4. A and B support HTTP-based FT according to former capability exchange. 5. A and B parameters: a. FT MAX SIZE A is larger than for B. b. Preferred File Transfer technology is HTTP. c. Standalone message enabled is set to false.
Test procedure	1. A selects one file from the local storage and then selects the file transfer option. 2. B receives the file download notification and sees an error message because of file size. 3. The file cannot be opened since it is not transferred to B.
Expected results **Post-conditions**	1. A sees that the file transfer is available for B. 2. B receives file download notification and sees the size of the file and a size error. A file preview may be shown. 3. The file is not transferred and received. 4. No display notification is received and shown on A.

Deep inspection	1. Verify that capability exchange, including FT, and HTTP FT tags, occurs before chat session. 2. Verify support of FT HTTP XML content type during the SDP negotiation. 3. Verify that FT notification XML is sent inside the active chat. 4. Verify that the URL complies to ftcontentserver.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org. 5. Depending on client implementation, verify (a) that no HTTPS requests for file and optional thumbnail are sent to the content server in A's network or (b) just an HTTPS request for the optional thumbnail is sent to the content server in A's network and its content is received by A. Note: It is assumed that the file will always upload successfully.

Test Case ID	**RCS_6_2_3_5**
Related test cases	**RCS_6_2_1_1**
Feature	File Transfer - File Transfer using HTTP
Purpose	Delayed HTTP-based file transfer in active one-to-one chat (temporary Content Server error or up/download race condition)
Pre-conditions Scenario	1. A and B are in an active one-to-one chat session. 2. B has enough free storage space to receive file. 3. File size being transferred is smaller than the warning file size (FT WARN SIZE). 4. A and B support HTTP-based FT according to former capability exchange. 5. File download is not possible initially, e.g., because file upload has not been completed but file download notification already sent or the content server in A's network is blocked for some time. 6. A and B parameters: a. FT MAX SIZE is the same for A and B. b. Preferred File Transfer technology is HTTP.
Test procedure	1. A selects one file from the local storage and then selects the file transfer option. 2. B receives and accepts the file download notification if prompted. 3. B opens file.
Expected results **Post-conditions**	1. A sees that the file transfer option is available for B. 2. B receives the file transfer download notification and sees the size of the file and an indication of the file type (e.g., MIME icon). A mini-preview, if available, is only shown after a delay. 3. The file and optional thumbnail are transferred and received after a delay and up to three automatic retries. 4. Display notification is received and shown on A.
Deep inspection	1. Verify that capability exchange, including HTTP FT tag, occurs before chat session. 2. Verify support of FT HTTP XML content type during SDP negotiation. 3. Verify that FT notification XML is sent inside the active chat. 4. Verify that the URL complies to ftcontentserver.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org.

	5. Verify that an HTTP error is sent back temporally as an answer to the HTTPS requests for file and optional thumbnail to the content server in A's network. 6. Depending on the error code sent by the content server verify (a) a retry after RetryAfter header interval out of HTTP 503 or (b) up to three retries in case of another HTTP error code. In case of unblocked server successful HTTP 200 responses with content to the last requests shall be received by the A. 7. Verify display notification in SIP MESSAGE. Note: It is assumed that the file will always upload successfully.

7 Messaging

7.1 *Standalone Messaging*

7.1.1 Message Processing

Test Case ID	**RCS_7_1_1_1**
Related Test Cases	
Feature	Standalone Messaging - Pager Mode
Purpose	Send Pager Mode message 1-to-1.
Pre-conditions Scenario	1. A and B are RCS Standalone Messaging capable. 2. A and B are shown as available.
Test procedure	A sends a small message (e.g., 100 characters) (Pager Mode Standalone Message) to B with delivery and display notifications requested.
Expected results **Post-conditions**	1. B receives the message. 2. The message sent by A is shown delivered if successful delivery. How soon the delivery status is available will depend on whether the message is delivered immediately or deferred.
Deep inspection	1. Verify that the message is delivered successfully, either immediately or deferred. The feature tag is ‘3gpp-service.ims.icsi.oma.cpm.msg’ for the immediate delivery and ‘3gpp-service.ims.icsi.oma.cpm.deferred’ for the deferred delivery. 2. Verify that the delivery notification is received (SIP MESSAGE).

Test Case ID	**RCS_7_1_1_2**
Related Test Cases	
Feature	Standalone Messaging - Pager Mode
Purpose	Send Pager Mode message one-to-many.
Pre-conditions Scenario	1. A, B and C are RCS Standalone Messaging capable. 2. A, B and C are shown as available.
Test procedure	A sends a small message (e.g., 100 bytes) (Pager Mode Standalone Message) to B and C with delivery and display notifications requested.
Expected results **Post-conditions**	1. B and C receive the message. 2. The message sent by A is shown as delivered if successful delivery was received from each recipient. How soon the delivery status is available will depend on whether the message is delivered immediately or deferred.
Deep inspection	1. Verify that the message is delivered successfully to each recipient, either immediately or deferred. The feature tag is '3gpp-service.ims.icsi.oma.cpm.msg' for immediate delivery and '3gpp-service.ims.icsi.oma.cpm.deferred' for the deferred delivery. 2. Verify that the delivery notification is received (SIP MESSAGE) from each recipient. 3. Verify that the recipient list history is sent across the NNI (in SIP message body).

Test Case ID	**RCS_7_1_1_3**
Related Test Cases	
Feature	Standalone Messaging - Large Message Mode
Purpose	Send Large Message Mode message one-to-one.
Pre-conditions Scenario	1. A and B are RCS Standalone Messaging capable. 2. A and B are shown as available.

Test procedure	1. A sends an image message (Large Message Mode Standalone Message) to B with delivery and display notifications requested. 2. B accepts the invitation (either auto or manual). 3. A sends a large text message (e.g., 1500 bytes) to B. 4. B accepts the invitation.
Expected results **Post-conditions**	1. B accepts the invitation (either auto or manual acceptance) and receives the image/text message. The multimedia attachment can be rendered and displayed. 2. The message sent by A is shown delivered if successful delivery. How soon the delivery status is available will depend on whether the message is delivered immediately or deferred.
Deep inspection	1. Verify that the invitation is received and accepted. 2. Verify that the message is delivered successfully, either immediately or deferred. The feature tag is '3gpp-service.ims.icsi.oma.cpm.largemsg' in SIP INVITE for the immediate delivery and '3gpp-service.ims.icsi.oma.cpm.deferred' for the deferred delivery. 3. Verify that the delivery notification is received (MSRP IMDN/CPIM)

Test Case ID	**RCS_7_1_1_4**
Related Test Cases	
Feature	Standalone Messaging - Large Message Mode
Purpose	Send Large Message Mode message one-to-many.
Pre-conditions Scenario	1. A, B and C are RCS Standalone Messaging capable. 2. A, B and C are shown as available.
Test procedure	1. A sends an image/large text message (Large Message Mode Standalone Message) to B and C with delivery and display notifications requested. 2. B and C accept the invitation (either auto-accept or manual).
Expected results **Post-conditions**	1. B and C receive the image/large text message. The multimedia attachment can be rendered and displayed. 2. The message sent by A is shown delivered if successful delivery to all recipients. How soon the delivery status is available will depend on whether the messages are delivered immediately or deferred.

Deep inspection	1. Verify that the invitation is received and accepted by each recipient. 2. Verify that the message is delivered successfully, either immediately or deferred. The feature tag is ‘3gpp-service.ims.icsi.oma.cpm.largemsg’ in SIP INVITE for the immediate delivery and ‘3gpp-service.ims.icsi.oma.cpm.deferred’ for the deferred delivery. 3. Verify that the delivery notification is received (MSRP CPIM) from each recipient. 4. Verify that the recipient list history is sent across NNI (in SIP INVITE).

7.1.2 Multi-device Handling

Test Case ID	**RCS_7_1_2_1**
Related Test Cases	**RCS_7_1_1_1**
Feature	Standalone Messaging - Pager Mode
Purpose	Send Pager Mode message from originating side with multi-device.
Pre-conditions Scenario	1. A and B are RCS Standalone Messaging capable. 2. A and B are shown as available. 3. A has two devices, A1 and A2, both online and active.
Test procedure	A1 device sends a small message (e.g., 100 bytes) (Pager Mode Standalone Message) to B with delivery and display notifications requested.
Expected results **Post-conditions**	1. B receives the message. 2. The message sent by A1 is shown delivered if successful delivery. How soon the delivery status is available will depend on whether the message is delivered immediately or deferred. 3. A2 will not receive any delivery notification. A2 will get the delivery message state by message synchronization if Network Message Storage is supported.
Deep inspection	Verify that the delivery notification is received (SIP MESSAGE) only on the device that originated the message.

Test Case ID	**RCS_7_1_2_2**
Related Test Cases	**RCS_7_1_1_1**

Feature	Standalone Messaging - Pager Mode
Purpose	Send Pager Mode message to a recipient with multiple devices.
Pre-conditions Scenario	1. A and B are RCS Standalone Messaging capable. 2. A and B are shown as available. 3. B has two devices, B1 and B2, both online and active.
Test procedure	1. A sends a small text message (Pager Mode Standalone Message) to B with delivery and display notification request. 2. Both B1 and B2 display the message.
Expected results **Post-conditions**	1. Both B1 and B2 receive the message. 2. A will receive only one delivery notification (and not multiple delivery notifications).
Deep inspection	1. Verify that the delivery notification is sent by B1 and B2. 2. Verify that the delivery notification is received only once (i.e., only one delivery notification will be sent across the NNI). The duplicate notification is discarded by B’s Messaging Server.

Test Case ID	**RCS_7_1_2_3**
Related Test Cases	**RCS_7_1_1_3**
Feature	Standalone Messaging - Large Message Mode
Purpose	Send Large Message Mode message from an originator with multi-device.
Pre-conditions Scenario	1. A and B are RCS Standalone Messaging capable. 2. A and B are shown as available. 3. A has two devices, A1 and A2, both online and active.
Test procedure	1. A1 device sends a multimedia message (Large Message Mode Standalone Message) to B with delivery and display notifications requested. 2. B accepts the invitation.

Expected results **Post-conditions**	1. B receives the message. 2. The message sent by A1 is shown delivered if successful delivery. 3. A2 will not receive any delivery notification. A2 will get the delivery message state by message synchronization if Network Message Store is supported.
Deep inspection	Verify that the delivery notification is received only on the device that originated the message.

Test Case ID	**RCS_7_1_2_4**
Related Test Cases	**RCS_7_1_1_3**
Feature	Standalone Messaging - Large Message Mode
Purpose	Send Large Message Mode message to a recipient with multi-device.
Pre-conditions Scenario	1. A and B are RCS Standalone Messaging capable. 2. A and B are shown as available. 3. B has two devices, B1 and B2, both online and active.
Test procedure	1. A sends a multimedia message (Large Message Mode Standalone Message) to B with delivery and display notifications requested. 2. Both B1 and B2 display the message.
Expected results **Post-conditions**	1. Both B1 and B2 receive the message. 2. A will receive one delivery notification only (and not multiple notifications).
Deep inspection	1. Verify that the delivery notification is sent by B1 and B2. 2. Verify that the delivery notification is received only once (i.e., only one delivery notification will be sent across NNI). The duplicate notification is discarded by B’s Messaging Server.

7.1.3 Exception Conditions

Test Case ID	**RCS_7_1_3_1**

Related Test Cases	
Feature	Standalone Messaging - Large Message Mode
Purpose	Maximum large message size exceeded.
Pre-conditions Scenario	1. A and B are RCS Standalone Messaging capable. 2. A and B are shown as available. 3. A NNI parameters: a. Max size of large message = xxx MB. 4. B NNI parameters: a. Max size of large message = yyy MB.
Test procedure	1. A sends a multimedia message (Large Message Mode Standalone Message) to B with the message size larger than B can support. 2. B's device sees that the file is too big and answers the INVITE with an error message. NOTE: If Network B's Messaging Server detects the file is too big, it will respond with an error message.
Expected results **Post-conditions**	1. B detects that the file to be sent is too big (i.e., a=file-selector value is too big). B answers the INVITE with a 488 "Not Acceptable Here" and warning header set to "133 Size exceeded". OR 1. B's Messaging Server detects that the file to be sent is too big (i.e., a=file-selector value is too big). B's Messaging Server answers the INVITE with a 403 "Forbidden" and warning header set to "133 Size exceeded". 2. A does not receive the delivery notification.
Deep inspection	1. Verify the a=file-selector attribute in the incoming SIP request. 2. Verify the Max size of large message parameter defined in Network B. 3. B's Messaging Server detects that the file to be sent is too big (i.e., a=file-selector value is too big) and responds the INVITE with a 403 "Forbidden" and warning header set to "133 Size exceeded". OR 1. B responds the INVITE with a 488 "Not Acceptable Here" and warning header set to "133 Size exceeded".

Test Case ID	**RCS_7_1_3_2**
Related Test Cases	
Feature	Standalone Messaging - Large Message Mode
Purpose	Unsupported Media Type
Pre-conditions Scenario	1. A and B are RCS Standalone Messaging capable. 2. A and B are shown as available.

Test procedure	A sends a multimedia message (Large Message Mode Standalone Message) to B with media type which is not supported by B.
Expected results **Post-conditions**	1. B accepts the INVITE so the session is set up between A and B. 2. B rejects the MSRP SEND request that contains a content-type not supported by B. 3. The session between A and B is torn down.
Deep inspection	Verify that the MSRP SEND response is set to 415.

7.2 1-to-1 Chat

7.2.1 Session Management

Test Case ID	**RCS_7_2_1_1**
Related Test Cases	
Feature	1-1 Chat - CPM NNI
Purpose	CPM user initiates an one-to-one chat with another CPM user.
Pre-conditions Scenario	1. A and B are OMA CPM users. 2. A and B are RCS Chat capable. 3. A and B are shown as available. 4. Network A NNI parameters: a. Message in INVITE =0 5. Network B NNI parameters: a. Message in INVITE =0
Test procedure	1. A initiates a one-to-one chat with B without typing a message (i.e., no message in SIP INVITE). 2. B is informed of the chat invitation and accepts the invitation (either auto or manual). 3. A and B are ready to chat.
Expected results **Post-conditions**	1. B accepts the invitation (either auto or manual acceptance). 2. Chat session is established and ready for the chat.

Deep inspection	1. Verify the invitation is received and accepted. 2. Verify the chat session is set up.

Test Case ID	**RCS_7_2_1_2**
Related Test Cases	
Feature	1-1 Chat - CPM NNI
Purpose	CPM user initiates a one-to-one chat with CPM user with first message in INVITE on originating network – accepted.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. A and B are RCS Chat capable. 4. A and B are shown as available. 5. Network A NNI parameters: a. Message in INVITE =1. Note: CPM NNI does not allow first message in INVITE.
Test procedure	1. A initiates chat with B by typing an initial message1 and then sends it to B. 2. B's device accepts the invitation (either auto or manual). 3. B receives the initial message1 displayed as a chat message.
Expected results **Post-conditions**	1. After B accepts, the chat session is set-up. 2. B receives message1. 3. A receives the delivery notification on the initial message(s), if requested. Note: Depending on how long it is before the acceptance is returned (e.g., network delay), the initial message may be received by B as Pager Mode standalone messages until the SIP INVITE is accepted (see Related Test Cases).
Deep inspection	1. Verify that the invitation is received and accepted. 2. Verify that the initial message is in the CPIM wrapper on the originating side. 3. If the terminating network supports Standalone Messaging and the session is not yet established, the initial message may be sent via Pager Mode Standalone Message. Verify if this is the case and that a delivery notification is sent by B and received by A via SIP MESSAGE. Verify that the chat session is set up and MSRP connection established. 4. If no Standalone Message is sent carrying the initial message, verify that the initial message is received and the delivery notification is sent via MSRP SEND as a chat message. 5. If a standalone message is received, (assuming supported by the receiving network), verify the Pager Mode standalone message is received with the same Conversation ID as the chat session.

Test Case ID	**RCS_7_2_1_3**
Related Test Cases	**RCS_7_2_1_2**
Feature	1-1 Chat - CPM NNI
Purpose	CPM user initiates a one-to-one chat with CPM user with first message in INVITE on originating network – delayed acceptance.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. A and B are RCS Chat capable. 4. B is Standalone Message capable. 5. A and B are shown as available. 6. Network A NNI parameters: a. Message in INVITE =1. Note: CPM NNI does not allow first message in INVITE.
Test procedure	1. A initiates chat with B by typing an initial message and then sends it to B. 2. B delays the acceptance of the invitation, but B will accept later before SIP INVITE times out. 3. B receives the initial message as a Pager Mode standalone message. 4. B accepts the invitation before SIP INVITE times out.
Expected results **Post-conditions**	1. Before B accepts the invitation, B receives the initial message as a Pager Mode standalone message. 2. After B accepts, the chat session is set-up. 3. A receives the delivery notification on the initial message, if requested.
Deep inspection	1. Verify that the initial message is in the CPIM wrapper in SIP INVITE from A. 2. Verify that the initial message is received by B as a Pager Mode Standalone Message. 3. Verify that A receives delivery notification for the initial message. 4. Verify that the chat session is set up after B accepts the SIP INVITE.

Test Case ID	**RCS_7_2_1_4**
Related Test Cases	
Feature	1-1 Chat - CPM NNI
Purpose	CPM user initiates a one-to-one chat with CPM user with first message in INVITE on originating network – no answer.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. A and B are RCS Chat capable. 4. A and B are shown as available. 5. Network A NNI parameters: a. Message in INVITE =1.

	6. Network B implements manual accept. Note: CPM NNI does not allow first message in INVITE.
Test procedure	1. A starts chat with B by typing an initial message and then sends it to B. 2. B will be informed of the incoming chat invitation. 3. B does not answer.
Expected results **Post-conditions**	1. B receives the initial message as a standalone message, if supported; otherwise, it is received as a legacy message. 2. A will receive the delivery notification for the initial message which is sent either as a standalone or legacy message.
Deep inspection	1. Verify that the invitation is received by B but a timeout occurs (SIP 408 or 487). Verify the initial message is received as either standalone message or legacy. 2. Verify A receives the delivery notification from B, if requested, for the initial message which is sent either as a standalone or legacy message.

Test Case ID	**RCS_7_2_1_5**
Related Test Cases	**RCS_7_2_1_2**
Feature	1-1 Chat - CPM NNI
Purpose	CPM user initiates a one-to-one chat with CPM user with first message in INVITE on originating network – rejected.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. A and B are RCS Chat capable. 4. A and B are shown as available. 5. Network A NNI parameters: a. Message in INVITE enabled. 6. CPM NNI does not allow first message in INVITE. 7. Network B implements manual accept.
Test procedure	1. A starts chat with B by typing an initial message and then sends it to B. 2. B will be informed of the incoming chat invitation. 3. B rejects the invitation.

Expected results **Post-conditions**	1. There is no chat session set-up up between A and B. 2. B does not receive the initial message.
Deep inspection	Verify that B rejects the invitation with 603, which is propagated back to A.

Test Case ID	**RCS_7_2_1_6**
Related Test Cases	
Feature	1-1 Chat - SIMPLE IM NNI
Purpose	SIMPLE IM user initiates a one-to-one chat with SIMPLE IM user with first message in INVITE - accepted.
Pre-conditions Scenario	1. A is a SIMPLE IM user. 2. B is a SIMPLE IM user. 3. A and B are RCS Chat capable. 4. A and B are shown as available. 5. Network A NNI parameters: a. Message in INVITE =1. 6. Network B NNI parameters: a. Message in INVITE =1. 7. Network A and Network B use Message Server.
Test procedure	1. A starts chat with B by typing an initial message and then sends it to B. 2. B gets the message (notification bar) and accepts the INVITE by responding to the message. 3. B sends the delivery notification for the initial message, if requested.
Expected results **Post-conditions**	1. B receives the initial message (notification bar). 2. When B starts to responds to the message, the chat session is set-up. 3. A receives the delivery notification to those initial messages sent before the chat session is set-up.
Deep inspection	1. Verify that the invitation is received. 2. Verify that the initial message is received. 3. Verify that A receives the delivery notification from B, if requested.

Test Case ID	**RCS_7_2_1_7**
Related Test Cases	**RCS_7_2_1_6**
Feature	1-1 Chat - SIMPLE IM NNI
Purpose	(IM-IM) SIMPLE IM initiates a one-to-one chat with SIMPLE IM user with first message in INVITE – no answer.
Pre-conditions Scenario	1. A is a SIMPLE IM user. 2. B is a SIMPLE IM user. 3. A and B are RCS Chat capable. 4. A and B are shown as available. 5. Network A NNI parameters: a. Message in INVITE =1. 6. Network B NNI parameters: a. Message in INVITE =1. 7. Network A and Network B use Message Server.
Test procedure	1. A starts chat with B by typing an initial message and sends it to B. 2. B gets the message, but does not respond. 3. A sends multiple messages (in multiple INVITE) with no response. 4. After some time, all INVITE requests expire, including the last INVITE.
Expected results **Post-conditions**	1. B sees the message from A in the notification bar but does not open the chat or respond to the message. 2. Chat session is not set up between A and B. 3. After the INVITE timeout, the INVITE is cancelled / terminated by Message Server. 4. A receives the delivery notification for those messages in INVITE requests.
Deep inspection	1. Verify that the initial messages are delivered via successful invitations. 2. Verify that the last invitation is cancelled (487) after invitation timeout (408), and Messaging Server maps the 408/487 into 486 before propagating it to the sender. Verify that B responds with 486 for each earlier INVITE when a new INVITE arrives. 3. Verify that the delivery notification to the initial message is received by A in the SIP message.

Test Case ID	**RCS_7_2_1_8**
Related Test Cases	
Feature	1-1 Chat - CPM NNI
Purpose	Simple IM initiates a one-to-one chat with CPM user with first message in INVITE – accepted.

Pre-conditions Scenario	1. A is a SIMPLE IM user. 2. B is a CPM Chat user. 3. A and B are RCS Chat capable. 4. A and B are shown as available. 5. Network A NNI parameters: a. Message in INVITE =1. 6. The interworking function implements a timer to wait for the session is setup for sending the initial message. Note: CPM NNI does not allow first message in INVITE.
Test procedure	1. A starts chat with B by typing an initial message and then sends it to B. 2. B is informed of the chat invitation. 3. B accepts the invitation (either auto or manual). B will receive the initial message displayed as a chat message. Note: Depending on how long it is before the response is received from the receiving network or the implementation of the interworking function, the initial and subsequent messages may be received by B as Pager Mode standalone message until the SIP INVITE is accepted.
Expected results **Post-conditions**	1. Until the SIP INVITE response from B is received, some of the initial messages may be received as Pager Mode standalone message(s). 2. After B accepts, the chat session is set-up between A and B. 3. A receives the delivery notification on the initial message(s), if requested.
Deep inspection	1. Verify that the invitation is received with the header mapping according to RCS NNI Interworking Guideline IR.90 section 6.3.1. 2. Verify that the initial message is received from the CPIM wrapper on the originating side (i.e., from IM user). 3. Verify that the chat session is set up and the MSRP connection is established, so the initial message is received and the delivery notification is sent in MSRP SEND. 4. Verify that a Pager Mode standalone message is received if so implemented by interworking function.

Test Case ID	**RCS_7_2_1_9**
Related Test Cases	**RCS_7_2_1_8**
Feature	1-1 Chat - CPM NNI
Purpose	SIMPLE IM user initiates a one-to-one chat with CPM user with first message in INVITE on originating network - no answer.
Pre-conditions Scenario	1. A is a SIMPLE IM user. 2. B is a CPM user. 3. A and B are RCS Chat capable. 4. A and B are shown as available. 5. Network A NNI parameters: a. Message in INVITE =1. 6. Network B implements manual accept.

	Note: CPM NNI does not allow first message in INVITE.
Test procedure	1. A starts chat with B by typing an initial message and then sends it to B. 2. B will be informed of the chat invitation. 3. B does not answer.
Expected results **Post-conditions**	1. B receives the initial message as standalone message, if supported; otherwise, it is received as a legacy message. 2. No chat session is set-up between A and B. 3. A receives the delivery notification for the initial message which is sent as either a standalone or legacy message).
Deep inspection	1. When the INVITE is timeout (SIP 408), verify that the initial message is received as either standalone message or legacy. 2. Verify that no chat session is setup. 3. Verify that A receives the delivery notification from B, if requested.

Test Case ID	**RCS_7_2_1_10**
Related Test Cases	**RCS_7_2_1_8**
Feature	1-1 Chat - CPM NNI
Purpose	SIMPLE IM user initiates a one-to-one chat with CPM user with first message in INVITE on originating network – rejected.
Pre-conditions Scenario	1. A is a SIMPLE IM user. 2. B is a CPM user. 3. A and B are RCS Chat capable. 4. A and B are shown as available. 5. Network A NNI parameters: a. Message in INVITE =1. 6. Network B implements manual accept. Note: CPM NNI does not allow first message in INVITE.
Test procedure	1. A starts chat with B by typing an initial message and then sends it to B. 2. B is informed of the chat invitation. 3. B rejects the invitation.
Expected results **Post-conditions**	1. B does not receive the initial message in the INVITE. 2. No chat session is setup between A and B. 3. A does not receive the delivery notification for the initial message.

Deep inspection	1. When the invite is rejected (SIP 603), verify that the initial message is not delivered to B. 2. Verify that no chat session is set-up. 3. Verify that A does not receive the delivery notification from B.

Test Case ID	**RCS_7_2_1_11**
Related Test Cases	
Feature	1-1 Chat - CPM NNI
Purpose	(CPM-IM) CPM user initiates a one-to-one chat with SIMPLE IM user – success.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a SIMPLE IM user. 3. A and B are RCS Chat capable. 4. A and B are shown as available. 5. Network B NNI parameters: a. Message in INVITE =1. Note: CPM NNI does not allow first message in INVITE.
Test procedure	1. A starts chat to B. 2. A types the message1. 3. B receives the initial message displayed as a chat message. 4. B responds to the message2. 5. Session is set-up between A and B.
Expected results **Post-conditions**	1. B receives the initial message (on notification bar) and the chat message1 from A. 2. A receives the delivery notification on message1, if requested. 3. When B responds to the message1, the chat session is set-up. 4. A receives the delivery notification on message2, if requested.
Deep inspection	1. Verify that the MSRP session is set-up on the A leg and messages sent by A are stored and copied to the CPIM body in the SIP INVITE request to B. 2. Verify that the SIP INVITE is received by B with a=accept-wrapped-types:text/plain. 3. Verify that the first message is included as the CPIM body in the SIP INVITE to B. 4. Verify that the delivery notification is received by A in the SIP MESSAGE for the first message1 and in the MSRP SEND for message2.

Test Case ID	**RCS_7_2_1_12**
Related Test Cases	**RCS_7_2_1_11**
Feature	1-1 Chat - CPM NNI

Purpose	(CPM-IM) CPM user initiates a one-to-one chat with SIMPLE IM user – no response or error response.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a SIMPLE IM user. 3. A and B are RCS Chat capable. 4. A and B are shown as available initially. 5. Network B NNI parameters: a. Message in INVITE =1. 6. Network B implements manual accept. Note: CPM NNI does not allow first message in INVITE.
Test procedure	1. A initiates chat with B. 2. A types a text message1. 3. Message1 is displayed on B’s device (e.g., notification bar). 4. No response is received from B (e.g., B is away from device). 5. Session is not set-up between A and B yet. 6. A types more messages message2 and message3. 7. Message2 and message3 are displayed on B’s device. 8. B still does not respond. 9. Session is still not set-up between A and B yet. 10. A few minutes later, B responds and a chat session setup. 11. A sends message4. 12. B receives message4.
Expected results **Post-conditions**	1. B receives chat message1, message2, and message3 (i.e., message displayed on chat window) but does not respond (e.g., ignores, away from device). 2. Later B accepts one invitation by responding to the message and a chat session is set up. 3. B receives messge4. 4. A receives a delivery notification, if requested, for each chat message that was delivered to B’s device.
Deep inspection	1. Verify that multiple SIP INVITE requests are received by B; one per each message sent by A until B responds to the INVITE (for this case, three SIP INVITE requests for three messages sent by A). 2. Verify that the latest session is used. That is, there is no multiple chat session with the same user at any time. B will send SIP BYE to cancel the previous session and accept the new INVITE to establish the new session. 3. Verify the delivery/display notifications for message1, message2, and message3 in the SIP INVITE are sent using the SIP MESSAGE by B and for message4 in the MSRP SEND.

Test Case ID	**RCS_7_2_1_13**
Related Test Cases	
Feature	1-1 Chat - CPM NNI
Purpose	One-to-One chat session timeout and restart.

Pre-conditions Scenario	1. A is a either a CPM or a SIMPLE IM user. 2. B is a either a CPM or a SIMPLE IM user. 3. A and B are RCS Chat capable. 4. A and B are shown as available. 5. Network A NNI parameter: a. One-to-one IM session inactivity timer = xxx sec. 6. Network B NNI parameter: a. One-to-one IM session inactivity timer = yyy sec. 7. The one-to-one chat session is active.
Test procedure	1. A is idle more than its one-to-one IM session inactivity timer. 2. A starts responding to a chat with B in the conversation and sends to B. 3. B accepts the invitation and receives the message according to the behaviour of the IM or CPM user.
Expected results **Post-conditions**	1. After no activity longer than A's IM session inactivity timer parameter, A starts chat which results in a new chat setup. 2. The chat session is started. 3. B receives messages.
Deep inspection	1. Verify that the session is timed out with a SIP BYE received by A and B. 2. Verify that the chat session is restarted with the SIP INVITE when one of the users starts a chat.

7.2.2 Message Handling

Test Case ID	**RCS_7_2_2_1**
Related Test Cases	**RCS_7_2_1_1**
Feature	1-1 Chat - CPM NNI
Purpose	CPM user sends a chat message to an CPM user.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. A and B are RCS Chat capable. 4. A and B are shown as available. 5. The one-to-one chat session between A and B is active. 6. Network A and Network B support multimedia contents in chat.
Test procedure	1. A sends a multimedia content, either from the gallery or camera. 2. B receives the multimedia message. 3. A sends a text message. 4. B receives messages and clicks on the multimedia content, and then the message is displayed / played.

Expected results **Post-conditions**	1. When A is composing, B receives an indication of A's typing. 2. B receives the messages. 3. A receives the delivery notifications, if requested.
Deep inspection	1. Verify that IsComposing notification is displayed while one side is typing. 2. Verify that the message is received by B. 3. Verify that A receives the delivery notification from B, if requested. 4. Verify that A receives the display notification from B, if enabled by B.

Test Case ID	**RCS_7_2_2_2**
Related Test Cases	**RCS_7_2_1_6**
Feature	1-1 Chat - IM NNI
Purpose	IM user sends a chat message to an IM user.
Pre-conditions Scenario	1. A is a SIMPLE IM user. 2. B is a SIMPLE IM user. 3. A and B are RCS Chat capable. 4. A and B are shown as available. 5. Network A NNI parameters: a. Multimedia in chat enabled is disabled. 6. Network B NNI parameters: a. Multimedia in chat enabled is disabled. 7. The one-to-one chat session between A and B is active.
Test procedure	1. A types a text message and sends it to B. 2. B receives the message. 3. B responds to the message.
Expected results **Post-conditions**	1. A is typing. B receives an indication of A's typing. A sends the message. 2. B will receive the message (notification bar) and read the message (open chat window). 3. A will receive the delivery notification and display notification, if enabled by B. 4. A will receive the message from B. 5. B will receive the delivery notification and display notification, if enabled by A.
Deep inspection	1. Verify that IsComposing notification is displayed while one side is typing. 2. Verify that the message is received by B. 3. Verify that A receives the delivery notification from B, if requested. 4. Verify that A receives the display notification from B, if enabled by B.

Test Case ID	**RCS_7_2_2_3**
Related Test Cases	**RCS_7_2_1_8**
Feature	1-1 Chat - CPM NNI
Purpose	CPM user sends chat messages.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a SIMPLE IM user. 3. A and B are RCS Chat capable. 4. A and B are shown as available. 5. Network A NNI parameters: a. Multimedia in chat enabled is enabled. 6. Network B NNI parameters: a. Multimedia in chat enabled is disabled. 7. Network A and Network B use Message Servers. 8. The one-to-one chat session is active.
Test procedure	1. A sends a text message1. 2. B receives the text message1 and responds with a text message2. 3. A sends a multimedia content message3.
Expected results **Post-conditions**	1. B receives the text message1. 2. B will receive the multimedia message3 as a Large Message Mode Standalone Message or as a File Transfer. 3. A will receive delivery notification for the multimedia message3, if it is sent as a Large Message Mode Standalone Message. A will not receive delivery notification if it is sent as a File Transfer.
Deep inspection	1. Verify that when B responds to the chat invitation, the a=accept-wrapped-types attribute did not contain any multimedia types. This will make it clear to A that to send a multimedia message, it has to be sent in a separate session as a Large Message Mode Standalone Message or as a File Transfer depending on local implementation. 2. Verify that the delivery notification is received by A for the text message1. 3. Verify that the delivery notification is received by A for the multimedia message, if it is sent as a Large Message Mode Standalone Message. A will not receive delivery notification if it is sent as a File Transfer.

Test Case ID	**RCS_7_2_2_4**
Related Test Cases	
Feature	1-1 Chat
Purpose	Verify disposition notifications sent when chat session is over.

Pre-conditions Scenario	1. A and B are either CPM or SIMPLE IM users. 2. A and B are RCS Chat capable. 3. A and B are shown as available. 4. The one-to-one chat session between A and B is active.
Test procedure	1. A types a message1 and sends the message. 2. B receives message1 and reads message1. 3. B is then away from the device. 4. A sends additional message2. 5. B receives message2 but does not read message2. 6. A and B are idle for some period time and the session is timeout. 7. B returns to the device and reads the message2.
Expected results **Post-conditions**	1. When A is composing, B receives an indication that A is typing. 2. When B reads messge1, A receives the delivery notification and display notification for message1. 3. A receives the delivery notification for message2. 4. At a later time, when B reads message2, A receives the display notification for message2.
Deep inspection	1. Verify that IsComposing notification is displayed while one side is typing. 2. Verify that delivery and display notifications for the message1 are carried in CPIM/IMDN in the MSRP SEND. 3. Verify that the display notification for message2 is in the SIP MESSAGE (because session is over).

7.2.3 Multi-device Handling

Test Case ID	**RCS_7_2_3_1**
Related Test Cases	**RCS_7_2_1_1, RCS_7_2_1_2**
Feature	1-1 Chat - CPM NNI
Purpose	CPM user initiates a one-to-one chat with multi-device user.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. A and B are RCS chat capable. 4. A and B are shown as available. 5. B has two devices, B1 and B2, both online / active. 6. Network A and Network B use Messaging Server. 7. Optionally, network-based Common Message Store is available in the Network B.
Test procedure	1. A initiates a chat with B. 2. B1 and B2 receive the invitation. 3. B2 accepts the invitation. 4. A sends message1. 5. B2 receives message1. 6. A sends additional messages: message2 and message3.

	7. B2 receives all messages. If network-based Common Message Store is available in the Network B, proceed with the following steps: 8. B1 displays all messages and the message state. 9. B1 sends message4. 10. A accepts the invitation. The chat session between A and B1 is set up. 11. A sends a message5. 12. B1 receives the message5. 13. B2 displays message4 and message5.
Expected results **Post-conditions**	1. The chat session is set up between A and B2 when B2 accepts the invitation. 2. The chat session towards B1 is terminated. 3. A receives the delivery notification for all messages sent. 4. The chat session is set-up between A and B1 when B1 sends message4. 5. A terminates the session between A and B2. 6. A receives message4. 7. B1 receives the delivery notification for message4 from A. 8. B1 receives message5. 9. A receives delivery notification for message5 from B1.
Deep inspection	1. Verify that both B1 and B2 receive the SIP INVITE from A. 2. Verify that the session toward B1 is terminated with the SIP BYE from the Messaging Server. 3. Verify that the delivery notification is received from B2 only. No multiple disposition notifications should be received or sent across NNI. B's Messaging Server will discard the duplicate delivery notification. 4. Verify that the session between A and B1 is active when B1 responds to message3 with message4 and the session between A and B2 is terminated.

Test Case ID	**RCS_7_2_3_2**
Related Test Cases	**RCS_7_2_1_6**
Feature	1-1 Chat IM NNI
Purpose	IM user with multi-device initiates one-to-one chat with IM user.
Pre-conditions Scenario	1. A is a SIMPLE IM user. 2. B is a SIMPLE IM user. 3. A and B are RCS chat capable. 4. A and B are shown as available. 5. A has two devices, A1 and A2, both online / active. 6. Network A and Network B use Messaging Server.
Test procedure	1. Device A1 initiates a chat with B. 2. Device A1 sends message1. 3. B receives message1. 4. A1 sends additional message2. 5. B receives all messages.

Expected results **Post-conditions**	1. The chat session is set-up between A1 and B. 2. A1 receives the delivery notifications for all messages.
Deep inspection	Verify that the delivery notification is received on A1 only.

7.2.4 Exception Conditions

Test Case ID	**RCS_7_2_4_1**
Related Test Cases	
Feature	1-1 Chat
Purpose	Chat message size limit
Pre-conditions Scenario	1. A and B are RCS chat capable. 2. A and B are shown as available. 3. Network A and Network B use Messaging Server. 4. Network A NNI Parameters a. Max size of 1-to-1 message = xx MB. 5. Network B NNI Parameters a. Max size of 1-to-1 message = yy MB. 6. Chat session is active.
Test procedure	A sends a message with size larger than what B can support.
Expected results **Post-conditions**	1. The Messaging Server B will not forward the message to B. 2. B does not receive the message. 3. A does not receive the delivery notification from B.
Deep inspection	1. Verify that B’s network detects the message is too large during transmission (MSRP SEND). The message is not delivered to B. 2. Verify that A receives MSRP failure REPORT and will not receive a delivery notification from B.

7.3 Group Chat

7.3.1 Session Management

Test Case ID	**RCS_7_3_1_1**
Related Test Cases	
Feature	Group Chat
Purpose	Initiate a group chat with dynamic focus.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. C is a SIMPLE IM user. 4. A, B, and C are RCS Chat capable. 5. A, B, and C are shown as available. 6. Network A NNI parameters: a. Auto-accept of group chat is enabled. b. Chat and group chat are enabled. c. Group Chat focus allowing restart is disabled (dynamic focus). 7. Network B NNI parameters: a. Auto-accept of group chat is enabled. b. Group chat full store and forward enabled. c. Group chat Invite only full store and forward enabled. d. Chat and group chat are enabled. 8. Network C NNI Parameters: a. Auto-accept of group chat is enabled. b. Chat and group chat are enabled.
Test procedure	1. A initiates group chat with B and C. 2. B receives the invitation and accepts the invitation. 3. Session is set-up with A and B. 4. A sends a message1. 5. C receives and accepts the invitation. 6. B receives the message1. 7. C receives the message1. 8. A receives a delivery notification from B and C. 9. A may receive a display notification from B or C.
Expected results **Post-conditions**	1. B receives and accepts the invitation. 2. Group session established with participant list: A (connected), B (connected), and C (pending). 3. C receives and accepts the invitation. 4. A and B get the notification of C having joined the conversation. Note: If B and C accept the invitation at close to the same time, the group chat will be set up with A, B, and C from the beginning and there would be no new notification event package. 5. B and C receive the message1. 6. A receives the delivery notification from B and C for message1.

Deep inspection	1. Verify that the conference state is sent when C joins. 2. Verify that the group chat session is set-up.

Test Case ID	**RCS_7_3_1_2**
Related Test Cases	**RCS_7_3_1_1**
Feature	Group Chat
Purpose	Initiate a group chat with static focus.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. C is a SIMPLE IM user. 4. A, B, and C are RCS Chat capable. 5. A, B, and C are shown as available. 6. Network A NNI parameters: a. Auto-accept of group chat is enabled. b. Chat and group chat are enabled. 7. Network B NNI parameters: a. Auto-accept of group chat is enabled. b. Group chat full store and forward is enabled. c. Group chat Invite only full store and forward is enabled. 8. Network C NNI parameters: a. Auto-accept of group chat is enabled. b. Group Chat focus allowing restart is enabled (static focus).
Test procedure	1. C initiates group chat with A and B. 2. A and B receive the invitation and accept the invitation. 3. Session is set-up with A and B. 4. C sends a message1. 5. A receives the message1. 6. B receives the message1. 7. C receives a delivery notification from A and B. 8. C may receive a display notification from A and B.
Expected results **Post-conditions**	1. Group session is established with participant list A, B and C. 2. When C sends a message1, A and B receive the message1. 3. C receives a delivery notification from A and B for the message1.
Deep inspection	1. Verify that the invitation is received by A and B with the header mapping according to RCS NNI Interworking Guideline IR.90 - section 6.3.1. 2. Verify that C receives a disposition notification for the message1 from A and B.

Test Case ID	**RCS_7_3_1_3**
Related Test Cases	
Feature	Group Chat
Purpose	Initiate a closed group chat.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. C is a SIMPLE IM user. 4. A, B, and C are RCS Chat capable. 5. A, B, and C are shown as available. 6. Network A NNI parameters: a. Auto-accept of group chat is enabled. 7. Network B NNI parameters: a. Auto-accept of group chat is enabled. b. Group chat full store and forward is enabled. c. Group chat Invite only full store and forward is enabled. 8. Network C is NNI Parameters: a. Auto-accept of group chat is enabled. b. Group Chat focus allowing restart is enabled.
Test procedure	1. B initiates a closed group chat with A and C. 2. A receives the invitation and accepts the invitation. 3. C receives the invitation and accepts the invitation. 4. A closed group session is set-up among A, B and C. 5. B sends a message1.
Expected results **Post-conditions**	1. The session invitation indicates the participants A, B, and C. 2. When A accepts the invitation, the group session is established with A and C. 3. When B accepts the invitation, A and C will get indication “B has joined the conversation” with the conference state. Note: If A and B accept the invitation near the same time, then the group session will be setup with A, B, and C from very beginning. 4. A receives the message1. 5. C receives the message1. 6. B receives a delivery notification from A and C. 7. B may receive a display notification from A or C.
Deep inspection	Verify that the attribute in the SDP of SIP INVITE contains the SDP attribute a=chatroom with OMA CPM 2.0 private token for a closed group chat.

Test Case ID	**RCS_7_3_1_4**
Related Test Cases	

Feature	Group Chat
Purpose	Group session timeout with dynamic focus.
Pre-conditions **Scenario**	1. A is a CPM user. 2. B is a CPM user. 3. C is a SIMPLE IM user. 4. A, B, and C are RCS Chat capable. 5. A, B, and C are shown as available. 6. Network A NNI parameters: a. Group IM session inactivity timer=xxx sec (e.g., 300). 7. Network B NNI parameters: a. Auto-accept of group chat is enabled. b. Group chat full store and forward is enabled. c. Group chat Invite only full store and forward is enabled. d. Group IM session inactivity timer=yyy sec (e.g., 300). 8. Network C NNI parameters: a. Auto-accept of group chat is enabled. b. Group Chat focus allowing restart is enabled. c. Group IM session inactivity timer=zzz sec (e.g., 300). 9. The group session is active and Network A is the conference focus with Group Chat.
Test procedure	1. B sends a message1. 2. A and C receive the messge1. 3. There is no activity for more than A's Group IM session inactivity timer and session timeout. 4. Session is torn down.
Expected results **Post-conditions**	It is transparent to A, B, and C when the session is terminated (because of timeout), and the session has been torn down.
Deep inspection	1. Verify that the focus server will terminate the subscription of all participants. 2. Verify that the SIP BYE 480 sent by the focus server C has a Reason header (e.g., SIP;cause=480;text="Bearer unavailable"). Note1: The Session Identity is not kept (Service Provider dependent) on the focus server. Note2: The latest participant list is not kept on the focus server.

Test Case ID	**RCS_7_3_1_5**
Related Test Cases	
Feature	Group Chat
Purpose	Group session timeout with static focus.

Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. C is a SIMPLE IM user. 4. A, B, and C are RCS Chat capable. 5. A, B, and C are shown as available. 6. Network A NNI parameters: a. Group IM session inactivity timer=xxx sec (e.g., 300). 7. Network B NNI parameters: a. Auto-accept of group chat is enabled. b. Group chat full store and forward is enabled. c. Group chat Invite only full store and forward is enabled. d. Group IM session inactivity timer=yyy sec (e.g., 300). 8. Network C NNI Parameters: a. Auto-accept of group chat is enabled. b. Group Chat focus allowing restart is enabled. c. Group IM session inactivity timer=zzz sec (e.g., 300). 9. The group session is active and Network C is the conference focus with Group Chat.
Test procedure	1. B sends a message1. 2. A and C receive the messge1. 3. There is no activity for more than C's Group IM session inactivity timer and session timeout. 4. Session is torn down.
Expected results **Post-conditions**	It is transparent to end user A, B and C when the session is terminated (because of timeout), and the session has been torn down.
Deep inspection	1. Verify that the focus server will terminate the subscription of all participants. 2. Verify that the SIP BYE 480 sent by the focus server C has a Reason header (e.g., SIP;cause=480;text="Bearer unavailable"). Note1: The Session Identity is kept (Service Provider dependent) on the focus server. Note2: The latest participant list is kept on the focus server.

Test Case ID	**RCS_7_3_1_6**
Related Test Cases	
Feature	Group Chat
Purpose	Restart group session with dynamic focus.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. C is a SIMPLE IM user. 4. A, B, and C are RCS Chat capable. 5. A, B, and C are shown as available.

	6. Network A NNI parameters: a. Auto-accept of group chat is enabled. b. Group IM session inactivity timer=xxx sec (e.g., 300). 7. Network B NNI parameters: a. Auto-accept of group chat is enabled. b. Group chat full store and forward is enabled. c. Group chat Invite only full store and forward is enabled. d. Group IM session inactivity timer=yyy sec (e.g., 300). 8. Network C NNI Parameters: a. Auto-accept of group chat is enabled. b. Group Chat focus allows restarts enabled. c. Group IM session inactivity timer=zzz sec (e.g., 300). 9. The group session is hosted by Network A with Group Chat (inactivity timeout); the last participant list has A, B and C.
Test procedure	1. C replies to the message1 on a conversation which has been idle for some time. 2. C types a message2 and sends it. 3. A and B receive an invitation from the focus server C with the previous Group Chat. 4. A and B accept the invitation. Group session is restarted with the focus server C.
Expected results **Post-conditions**	1. It is transparent to A, B, and C as to when the session is restarted. 2. A and B receive the message2. 3. C receives a delivery notification from A and B.
Deep inspection	1. Verify that the rejoin INVITE from C with Session Identity A will fail with 404. 2. Verify that C initiates a new group session with the same Group Chat and new Session Identity C using the last participant list from the last conference event package notification.

Test Case ID	**RCS_7_3_1_7**
Related Test Cases	
Feature	Group Chat
Purpose	Restart group session with static focus.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. C is a SIMPLE IM user. 4. A, B, and C are RCS Chat capable. 5. A, B, and C are shown as available. 6. Network A NNI parameters: a. Auto-accept of group chat is enabled. b. Group IM session inactivity timer=xxx sec (e.g., 300). 7. Network B NNI parameters:

	a. Auto-accept of group chat is enabled. b. Group chat full store and forward is enabled. c. Group chat Invite only full store and forward is enabled. d. Group IM session inactivity timer=yyy sec (e.g., 300). 8. Network C NNI parameters: a. Auto-accept of group chat is enabled. b. Group Chat focus allowing restart is enabled. c. Group IM session inactivity timer=zzz sec (e.g., 300). 9. The group session hosted by Network C with Group Chat (inactivity timeout); the last participant list has A, B, and C.
Test procedure	1. A types a message2 on a conversation (hosted by the focus server C). 2. B and C receive invitation from the focus server C with the previous Group Chat. 3. B and C accept the invitation. 4. Group session restarted (with the same focus server C).
Expected results **Post-conditions**	1. It is transparent to A, B, and C as to when the session is restarted. 2. B and C receive the message2 (session restarted). 3. A receives a delivery notification.
Deep inspection	1. Verify that the rejoin INVITE from A with Session Identity C will invoke the focus server C to initiate a group session with the last participant list (having A, B, and C) which is kept by the focus server C. 2. Verify that C initiates the group session with the same Group Chat test case ID using the last participant list stored when the session was timeout due to inactivity.

Test Case ID	**RCS_7_3_1_8**
Related Test Cases	
Feature	Group Chat
Purpose	Restart a closed group session.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. C is a SIMPLE IM user. 4. A, B, and C are RCS Chat capable. 5. A, B, and C are shown as available. 6. Network A NNI parameters: a. Group IM session inactivity timer=xxx sec (e.g., 300). 7. Network B NNI parameters: a. Auto-accept of group chat is enabled. b. Group chat full store and forward is enabled. c. Group chat Invite only full store and forward is enabled. d. Group IM session inactivity timer=yyy sec (e.g., 300). 8. Network C NNI parameters: a. Auto-accept of group chat is enabled. b. Group Chat focus allowing restart is enabled. c. Group IM session inactivity timer=zzz sec (e.g., 300).

	9. The closed group session is idle and Network B is the conference focus with Group Chat. 10. A is Standalone Message capable. 11. B is Standalone Message capable. 12. C is not Standalone Message capable.
Test procedure	1. A types a message3 on an idle chat where B and C are on the participant list (hosted by B). 2. B sends a message4.
Expected results **Post-conditions**	1. B and C receive session an invitation (for a closed group chat) and accept it. 2. B and C receive the message3. 3. A receives a delivery notification from B and C. 4. A and C receive the message4. 5. B receives a delivery notification from A and C.
Deep inspection	1. Verify that the restart INVITE from A with Session Identity B fails with 404 (due to dynamic focus). 2. Verify that the attribute a=chatroom with private token for CPM 2.0 closed group chat in the SDP of SIP INVITE request from A.

Test Case ID	**RCS_7_3_1_9**
Related Test Cases	
Feature	Group Chat
Purpose	Add new participant to a closed group chat.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. C is a SIMPLE IM user. 4. A, B, and C are RCS Chat capable. 5. A, B, and C are shown as available. 6. Network A NNI parameters: a. Group IM session inactivity timer=xxx sec (e.g., 300). 7. Network B NNI parameters: a. Auto-accept of group chat is enabled. b. Group chat full store and forward enabled. c. Group chat Invite only full store and forward is enabled. d. Group IM session inactivity timer=yyy sec (e.g., 300). 8. Network C NNI parameters: a. Auto-accept of group chat is enabled. b. Group Chat focus allowing restart is enabled. c. Group IM session inactivity timer=zzz sec (e.g., 300). 9. The closed group session is idle and Network B is the conference focus with Group Chat. 10. A is Standalone Message capable. 11. B is Standalone Message capable. 12. C is not Standalone Message capable.

Test procedure	A adds D to the group chat.
Expected results **Post-conditions**	Adding D is denied. (UI dependent: A may get a warning message: “No new participant can be added to a closed group chat”.)
Deep inspection	Verify that A receives a SIP 403 “Forbidden” response with Warning header text set to “127 Service not authorised”

Test Case ID	**RCS_7_3_1_10**
Related Test Cases	
Feature	Group Chat
Purpose	Restart group session from full store and forward participant.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. C is a SIMPLE IM user. 4. A, B, and C are RCS Chat capable. 5. A, B, and C are shown as available. 6. Network A NNI parameters: a. Group IM session inactivity timer=xxx sec (e.g., 300). 7. Network B NNI parameters: a. Auto-accept of group chat is enabled. b. Group chat full store and forward is enabled. c. Group chat Invite only full store and forward is enabled. d. Group IM session inactivity timer=yyy sec (e.g., 300). 8. Network C NNI parameters: a. Auto-accept of group chat is enabled. b. Group Chat focus allowing restart is enabled. c. Group IM session inactivity timer=zzz sec (e.g., 300). 9. The group session is not active and Network A is the conference focus with Group Chat. 10. A is Standalone Message capable. 11. B is Standalone Message capable. 12. C is not Standalone Message capable.
Test procedure	B types a message3 on an idle conversation where A and C are on the participant list (hosted by A).
Expected results **Post-**	1. A receives a session invitation (for a Large Message Mode message) and accepts it. 2. A receives the message3. 3. C receives the message3 as legacy (SMS/MMS) message.

conditions	4. No group chat session is set-up. 5. B receives a delivery notification from A and C. Note: The expected result is implementation dependent. The use case here is that Network B will not initiate the group chat due to theto the configuration of "Group chat Invite only full store and forward is enabled". Instead, NeworkNetwork B will send a standalone message (if the recipient is standalone message capable or legacy message if the recipient is not standalone message capable. Whether sending Pager or Large Message Mode or SMS or MMS, will depend on the message size and content type.
Deep inspection	1. Verify that the rejoin INVITE from B with Session Identity A will fail with 404. 2. Verify that no group chat session will be initiated by B 3. Verify that a Large Message mode session invitation is initiated to A (but not to C). 4. Verify that legacy message is sent either via SMS/MMS Interworking or from device client directly based on capability. Note: the deep inspection result is implementation dependent

Test Case ID	**RCS_7_3_1_11**
Related Test Cases	
Feature	Group Chat
Purpose	Restart group session with one user offline or no answer.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. C is a SIMPLE IM user. 4. A, B, and C are RCS Chat capable. 5. A, B, and C are shown as available. 6. Network A NNI parameters: a. Group IM session inactivity timer=xxx sec (e.g., 300). 7. Network B NNI parameters: a. Auto-accept of group chat is enabled. b. Group chat full store and forward is enabled. c. Group chat Invite only full store and forward is enabled. d. Group IM session inactivity timer=yyy sec (e.g., 300). 8. Network C NNI parameters: a. Auto-accept of group chat is enabled. b. Group Chat focus allowing restart is enabled. c. Group IM session inactivity timer=zzz sec (e.g., 300). 9. A is Standalone Message capable. 10. B is Standalone Message capable. 11. C is not Standalone Message capable. 12. The group chat hosted by C is active.

Test procedure	1. A, B and C are in chat session and then stop exchanging messages for a while. B goes offline. Then the group session times out. 2. C sends a message1 and starts exchanging messages. 3. A receives an invite and accepts it. 4. B is still offline. 5. A gets the message1. 6. B goes online and rejoins the group chat session. 7. A and C get indication that B has joined. 8. A sends a message2. 9. B and C receive the message2.
Expected results **Post-conditions**	1. When C sends the message1, the group session restarts with A, B, and C. 2. The invite to B times out and A accepts the invite. 3. When B comes online, B is able to re-join the group chat because of static focus. A and C are informed of B's having joined. 4. When C sends the message2, both A and B receive the message2.
Deep inspection	Verify that the invite is sent to B and C using previous Group Chat test case ID. Verify that B sends re-join INVITE to the same session identity of C.

Test Case ID	**RCS_7_3_1_12**
Related Test Cases	
Feature	Group Chat
Purpose	Extend 1-1 chat to group chat
Pre-conditions **Scenario**	1. A is a CPM user 2. B is a CPM user 3. C is a SIMPLE IM user or CPM user 4. A, B and C are RCS Chat capable 5. A, B and C are shown as available 6. Network A NNI parameters: a. Group IM session inactivity timer=xxx sec (e.g., 300) 7. Network B NNI parameters: a. Auto-accept of group chat enabled b. Group chat full store and forward enabled c. Group chat Invite only full store forward enabled d. Group IM session inactivity timer=yyy sec (e.g., 300) 8. Network C NNI Parameters: a. Auto-accept of group chat enabled b. Group Chat focus allows restarts enabled c. Group IM session inactivity timer=zzz sec (e.g., 300) d. Multimedia in chat enabled is disabled 9. A is StandaloneMsg capable 10. B is StandaloneMsg capable

	11. C is not StandaloneMsg capable 12. The 1-1 chat session between A and C is active
Test procedure	1. A and C are in ongoing chat session 2. A add B to the chat 3. B receives the invite and accepts it 4. C also receives the invite and accept it 5. C sends text message1 6. Both A and B get message1 7. B sends message2 and both A and C get message2
Expected results **Post-conditions**	1. When the new group chat is created, C receives the group chat invite. C may get an indication that the group chat is a replacement for the 1-1 chat between A and C 2. C accepts it, and tears down the 1-1 chat between A and C. 3. B receives the group chat invite 4. B accepts it 5. When C sends text message1, both A and B get the message1 6. When B sends text message2, both A and C get the message2
Deep inspection	1. Verify that the SIP INVITE to C has the Session-Replaces header so C may indicate to user C that this is a replacement for the 1-1 chat 2. Verify that C subscribes to the conference state event package to get the participant to display the participant list to user C 3. Verify that B subscribes to the conference state event package to get the participant to display the participant list to user B

Test Case ID	**RCS_7_3_1_13**
Related Test Cases	
Feature	Group Chat
Purpose	Leaving a group chat.
Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. C is a SIMPLE IM user. 4. A, B, and C are RCS Chat capable. 5. A, B, and C are shown as available. 6. Network A NNI parameters: a. Group IM session inactivity timer=xxx sec (e.g., 300). 7. Network B NNI parameters: a. Auto-accept of group chat is enabled. b. Group chat full store and forward is enabled. c. Group chat Invite only full store and forward is enabled. d. Group IM session inactivity timer=yyy sec (e.g., 300). 8. Network C NNI Parameters:

	a. Auto-accept of group chat enabled. b. “Multimedia in chat enabled” is disabled. c. Group Chat focus allowing restart is enabled. d. Group IM session inactivity timer=zzz sec (e.g., 300). 9. A is Standalone Message capable. 10. B is Standalone Message capable. 11. C is not Standalone Message capable. 12. The regular group chat hosted by C is active.
Test procedure	A leaves the group chat.
Expected results **Post-conditions**	B and C receive notification that A has left.
Deep inspection	1. Verify that A sends SIP BYE to the Conference focus server C. 2. Verify that conference state event package notification is sent to B and C. 3. Verify that Conference focus server C removes A from the participant list.

Test Case ID	**RCS_7_3_1_14**
Related Test Cases	
Feature	Group Chat
Purpose	User rejoins an inactive group session that the user has left voluntarily.

Pre-conditions Scenario	1. A is a CPM user. 2. B is a CPM user. 3. C is a SIMPLE IM user. 4. A, B, and C are RCS Chat capable. 5. A, B, and C are shown as available. 6. Network A NNI parameters: a. Group IM session inactivity timer=xxx sec (e.g., 300). 7. Network B NNI parameters: a. Auto-accept of group chat is enabled. b. Group chat full store and forward is enabled. c. Group chat Invite only full store and forward is enabled. d. Group IM session inactivity timer=yyy sec (e.g., 300). 8. Network C NNI parameters: a. Auto-accept of group chat is enabled. b. “Multimedia in chat enabled” is disabled. c. Group Chat focus allowing restart is enabled. d. Group IM session inactivity timer=zzz sec (e.g., 300). 9. A is Standalone Message capable. 10. B is Standalone Message capable. 11. C is not Standalone Message capable. 12. A has left the group chat hosted by C previously. 13. The regular group chat hosted by C is not active.
Test procedure	A tries to rejoin by responding a message to the group chat hosted by C that A left voluntarily.
Expected results **Post-conditions**	The attempt to rejoin fails because of authorization.
Deep inspection	1. Verify that the Session Identity C and Group Chat ID C are included in the rejoin INVITE from A and the rejoin fails with 403. 2. Verify that the rejoin from A who has left explicitly will fail with 404 while the group chat hosted by C is inactive.

Test Case ID	**RCS_7_3_1_15**
Related Test Cases	
Feature	Group Chat

Purpose	Rejoin to an active regular group chat that the user has left previously.
Pre-conditions **Scenario**	1. A is a CPM user. 2. B is a CPM user. 3. C is a SIMPLE IM user. 4. A, B, and C are RCS Chat capable. 5. A, B, and C are shown as available. 6. Network A NNI parameters: a. Group IM session inactivity timer=xxx sec (e.g., 300). 7. Network B NNI parameters: a. Auto-accept of group chat is enabled. b. Group chat full store and forward is enabled. c. Group chat Invite only full store and forward is enabled. d. Group IM session inactivity timer=yyy sec (e.g., 300). 8. Network C NNI parameters: a. Auto-accept of group chat is enabled. b. “Multimedia in chat enabled” is disabled. c. Group Chat focus allowing restart is enabled. d. Group IM session inactivity timer=zzz sec (e.g., 300). 9. A is Standalone Message capable. 10. B is Standalone Message capable. 11. C is not Standalone Message capable. 12. A has left group chat hosted by C. 13. The regular group chat hosted by C is active with participants B and C.
Test procedure	1. A tries to rejoin the group chat session with the Session Identity C that A left previously by responding to a chat message from the group chat. 2. B adds A to the group chat hosted by C. 3. C is notified that A has joined. 4. C sends a message1.
Expected results **Post-conditions**	1. When A tries to rejoin, the rejoin fails with an SIP 403 “Forbidden” response and a warning text set to “122 Function not allowed” in the response. 2. When B adds A, A successfully joins the active group chat hosted by C. 3. B and C are notified of A’s having joined and being listed in the participant list. 4. When C sends the message1, A and B receive the message1.
Deep inspection	1. Verify that when A tries to rejoin the chat group, a SIP 403 “Forbidden” response and a Warning header with the warning text set to “122 Function not allowed” are returned. 2. Verify that when B adds A, the SIP REFER with 202 response is returned.

7.3.2 Message Handling

No tests

7.3.3 Multi-device Handling

The test case for Multi-device handling for group chat is the same as the one for one-to-one chat.

7.3.4 Exception Conditions

Test Case ID	**RCS_7_3_4_1**
Related Test Cases	
Feature	Group Chat
Purpose	Maximum size of IM content limit.
Pre-conditions Scenario	1. A, B, and C are RCS chat capable. 2. A, B, and C are shown as available. 3. Network A, Network B and Network C use Messaging Server. 4. Chat session is active. 5. Network A NNI parameters: a. Maximum size of IM content = xx MB. 6. Network B NNI parameters: a. Maximum size of IM content = yy MB. 7. Network C NNI parameters: a. Maximum size of IM content = zz MB. 8. The group session hosted by Network A is active.
Test procedure	C sends a message that is larger than what B can support.
Expected results **Post-conditions**	1. A receives the message. 2. B does not receive the message. 3. C receives the delivery notification from A but does not receive the delivery notification from B.
Deep inspection	1. If the Conference Server A receives the “max-size” attribute of SDP from B, the Conference server A will not forward the message to B. Note: If the Conference Server A does not receive the “max-size” attribute of SDP from B, it sends the message to B Network. If B’s Messaging Server detects (from the total size of the message in the Byte-Range parameter of MSRP SEND) the message exceeds its “Maximum size of IM content”, it will not forward the message to B. 2. Verify that C does not receive the delivery notification from B.

Test Case ID	**RCS_7_3_4_2**
Related Test Cases	
Feature	Group Chat
Purpose	Maximum number of participants in group chat.

Pre-conditions Scenario	1. A, B, and C are RCS chat capable. 2. A, B, and C are shown as available. 3. Network A, Network B, and Network C use Messaging Server, 4. Network A NNI parameters: a. Maximum number of participants in group chat session = xx. 5. Network B NNI parameters: b. Maximum number of participants in group chat session = yy. 6. Network C NNI parameters: c. Maximum number of participants in group chat session = zz. 7. The group session hosted by Network A is active.
Test procedure	C adds a new participant that will make the number of participants to exceed A's "Maximum number of participants in group chat session".
Expected results **Post-conditions**	1. Adding of a participant fails. No new participant is added. 2. C may receive an error message on UI (UI dependent).
Deep inspection	1. Verify that the maximum number of participants allowed and the current user count for the running group chat is notified by the focus in the maximum-user-count and user-count elements. 2. If the maximum-user-count and user-count elements for the running group chat is notified by the focus in or it is provided in the notification event package but is not honoured by C, C will receive a SIP 486 "Busy Here" response and Warning header with the warning text set to "102 Too many participants".

8 Content Sharing

8.1 Video Share

8.1.1 Basic Video Share

Test case ID	**RCS_8_1_1_1**
Related test cases	
Feature	CONTENT SHARING - Video Share
Purpose	A shares video with B.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Video Share is possible. 3. Both devices have front or rear facing camera.

Test procedure	1. A starts Video Share to B. 2. B accepts the invite. 3. A sends live video to B.
Expected results **Post-conditions**	B receives the video.
Deep inspection	1. Verify share session establishment (including SDP negotiation). 2. Verify graceful termination of the share session (signalling) via traces. 3. Verify graceful termination of the share session (media) via traces. 4. If a video session, verify H.264 support in SDP negotiation.

Test case ID	**RCS_8_1_1_2**
Related test cases	**RCS_8_1_1_1**
Feature	CONTENT SHARING - Video Share
Purpose	Video Share: Successful simultaneous invites.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Video Share is possible. 3. Both handsets support simultaneous Video Share. 4. Both devices have front or rear facing camera.
Test procedure	1. A sends B Video Share invite. 2. The procedure is then repeated in opposite order.
Expected results **Post-conditions**	Two Video Share sessions are established (A->B and B->A).
Deep inspection	1. Verify graceful establishment of the Video Share session (signalling) via traces (both sessions). 2. Verify graceful establishment of the Video Share session (media) via traces (both sessions). 3. Verify H.264 support in SDP negotiation. 4. Verify capability exchange.

Test case ID	**RCS_8_1_1_3**
Related test cases	
Feature	CONTENT SHARING - Video Share
Purpose	Simultaneous Video and Image Share sessions
Pre-conditions Scenario	1. A and B are in an active voice call. 2. There is an active Video Share session from A to B. 3. Both handsets support simultaneous Video/Image Share sessions (one per direction). 4. Both devices have front or rear facing camera.
Test procedure	1. A is sharing video with B. Then B decides to share a picture with A. 2. The procedure is then repeated in opposite order.
Expected results **Post-conditions**	1. The Image Share transference is completed. A can see the image that B sent while B can see the video that A sent. 2. Each share session is independent and should be handled separately.
Deep inspection	1. Verify graceful establishment of the Image Share session (signalling) via traces (active session) and cancellation of the other invitation. 2. Verify graceful establishment of the Image Share session (media) via traces (active session). 3. Verify graceful completion of Image Share session and signalling/media release. 4. Verify capability exchange.

Test case ID	**RCS_8_1_1_4**
Related test cases	**RCS_8_1_1_1**
Feature	CONTENT SHARING - Video Share
Purpose	Resilience to several share sessions start and completion
Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Video Share is possible. 3. Both devices have front or rear facing camera.

Test procedure	A is able to share several items [1). start and finish a Video Share; 2). then share an image; 3). Then finally another image] one after the other during the same call with B.
Expected results **Post-conditions**	The Video/Image Share are performed according to the specification.
Deep inspection	1. Verify graceful initiation/termination of the different sessions (signalling) via traces. 2. Verify graceful handling of media during the sessions via traces.

Test case ID	**RCS_8_1_1_5**
Related test cases	
Feature	CONTENT SHARING - Video Share
Purpose	MSISDN normalization (on outgoing call) provided by network
Pre-conditions Scenario	
Test procedure	A establishes a voice call to B.
Expected results **Post-conditions**	1. B receives a call invitation with a number in the international format (E.164). 2. The outgoing voice call from A to B is successfully established.
Deep inspection	Verify MSISDNs formats in outgoing requests, including tel-uri with phone-context=domain.

Test case ID	**RCS_8_1_1_6**
Related test cases	**RCS_8_1_1_5**
Feature	CONTENT SHARING - Video Share
Purpose	Identification of the number against the local phone book
Pre-conditions Scenario	Only the local format for the number of B is known (e.g., prefix 0).
Test procedure	1. B establishes a voice call to A. 2. A receives a call invitation with a number in the international format. 3. On A's device, inspection of the phone book is done and B's contact is successfully identified.
Expected results **Post-conditions**	The incoming voice call from B to A is successfully established.
Deep inspection	Verify MSISDNs formats in incoming requests and responses, including sip-uri with user=phone and phone-context=domain, or tel-uri with phone-context=domain.

8.1.2 Multi-party call and Video Share

Test case ID	**RCS_8_1_2_1**
Related test cases	
Feature	CONTENT SHARING - Video Share
Purpose	Multi-party. Users were in a CS call without using the image or Video Share services.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Video Share is possible. 3. Both devices have front or rear facing camera.

Test procedure	1. A establishes a Video Share session with B. 2. A establishes a multi-party call with B and C. 3. C leaves the call so the call becomes a standard call between A and B.
Expected results **Post-conditions**	1. Voice call established between A and B. 2. Multi-party call is established among A, B, and C. 3. Video sharing between A and B ends. 4. RCS capabilities are updated for A. 5. Multiparty call has ended. 6. Both Image and Video Share capabilities are shown as available for A.
Deep inspection	1. Verify graceful establishment of the Video Share session (signalling) via traces (active session) and cancellation of the other invitation. 2. Verify graceful establishment of the Video Share session (media) via traces (active session). 3. Verify H.264 support in SDP negotiation. 4. Verify session termination (SIP/media). 5. Verify capability exchange following multiparty call establishment. 6. Verify capability exchange following multi-party call release.

8.1.3 Call hold and Video Share

Test case ID	**RCS_8_1_3_1**
Related test cases	
Feature	CONTENT SHARING - Video Share
Purpose	Call on hold. Users were in a CS call using Video Share.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Video Share is possible. 3. Both devices have front or rear facing camera.
Test procedure	1. A establishes a voice call with B. 2. A invites B to share live video. 3. B accepts Video Share. 4. A puts B’s voice call on hold. 5. A retrieves B’s voice call.
Expected results **Post-conditions**	1. Voice call established is between A and B. 2. Video sharing starts between A and B. 3. Voice call is placed on hold. 4. Video sharing between A and B ends. 5. RCS capabilities are updated for A resulting in image and Video Share services no longer available. 6. Voice connection is re-established between A and B.

	7. Video Share is re-established.
Deep inspection	1. Verify graceful establishment of the Video Share session (signalling) via traces (active session) and cancellation of the other invitation. 2. Verify graceful establishment of the Video Share session (media) via traces (active session). 3. Verify H.264 support in SDP negotiation. 4. Verify session termination (sip/media). 5. Verify capability exchange following the call on hold action does NOT take place. 6. Verify re-establishment of Video Share session.

Test case ID	**RCS_8_1_3_2**
Related test cases	
Feature	CONTENT SHARING - Video Share
Purpose	Call on hold. A performs video sharing with C, after placing a call with B on hold.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Video Share is possible.
Test procedure	1. A establishes a voice call with B. 2. A invites B to share a stored video. 3. B accepts video sharing. 4. A puts B's call on hold. 5. A establishes a voice call with C. 6. A invites C to share a stored video. 7. C accepts video sharing.
Expected results **Post-conditions**	1. Voice call established between A and B. 2. Video sharing starts A and B. 3. Voice call between A and B is placed on hold. 4. Picture sharing between A and B ends. 5. RCS capabilities are updated for A resulting in image and Video Share services no longer available. 6. Voice call is established between A and C. 7. Video sharing starts between A and C.
Deep inspection	1. Verify graceful establishment of the Video Share session (signalling) via traces (active session) and cancellation of the other invitation. 2. Verify graceful establishment of the Video Share session (media) via traces (active session). 3. Verify H.264 support in SDP negotiation.

	4. Verify session termination (sip/media). 5. Verify capability exchange following call on hold establishment (with B) does not take place but it does with C (call become active. 6. Verify 1-4 for new session.

Test case ID	**RCS_8_1_3_3**
Related test cases	
Feature	CONTENT SHARING - Video Share
Purpose	Call on hold. Retrieve call on hold + initiate Video Share.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Video Share is possible.
Test procedure	1. A establishes a voice call with B. 2. A invites B to share a stored video. 3. B accepts video sharing. 4. A puts a call with B on hold. 5. A retrieves B’s voice call. 6. A invites B to share a stored video. 7. B accepts video sharing.
Expected results **Post-conditions**	1. Voice call is established between A and B. 2. Video sharing starts between A and B. 3. Voice call between A and B is placed on hold. 4. Video sharing between A and B ends. 5. RCS capabilities are updated for A resulting in image and Video Share services no longer available. 6. Voice call between A and B is retrieved. 7. Video sharing starts.
Deep inspection	1. Verify graceful establishment of the Video Share session (signalling) via traces (active session) and cancellation of the other invitation. 2. Verify graceful establishment of the Video Share session (media) via traces (active session). 3. Verify H.264 support in SDP negotiation. 4. Verify session termination (SIP/media). 5. Verify capability exchange following a call on hold establishment (with B) and with B again when the call becomes active. 6. Verify 1-4 for new session.

8.1.4 Call waiting and Video Share

Test case ID	**RCS_8_1_4_1**
Related test cases	
Feature	CONTENT SHARING - Video Share
Purpose	Call waiting. Users were in a CS call using Video Share.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Video Share is possible.
Test procedure	1. A establishes a voice call with B. 2. A invites B to share a stored video. 3. B accepts the video sharing. 4. C calls A. A doesn't take the C’s call.
Expected results **Post-conditions**	1. Voice call is established between A and B. 2. Video sharing between A and B starts. 3. Video sharing between A and B continues.
Deep inspection	Verify that there is no impact from the call waiting on to the current session.

8.1.5 Multi-device handling

Test case ID	**RCS_8_1_5_1**
Related test cases	
Feature	CONTENT SHARING - Video Share
Purpose	A shares video with B; B is using two devices.
Pre-conditions Scenario	1. B has two devices, B1 and B2, and both are online and active. 2. A and B are in an active voice call; B is using his B1 device. 3. Capabilities exchange has taken place confirming Video Share is possible. 4. Both devices have front or rear facing camera.

Test procedure	1. After the voice call is established, A starts Video Share to B. 2. B accepts the invitation using his B2 device. 3. A sends video to B; Video can be a pre-stored (stream) or live (camera[s]).
Expected results **Post-conditions**	1. B receives the video at his B2 device. 2. No more sharing capabilities available for A.
Deep inspection	1. Verify share session establishment (including INVITE forking and SDP negotiation). 2. Verify graceful termination of the share session (signalling) via traces. 3. Verify graceful termination of the share session (media) via traces. 4. Verify H.264 support in SDP negotiation.

8.1.6 Exception conditions

Test case ID	**RCS_8_1_6_1**
Related test cases	
Feature	CONTENT SHARING - Video Share
Purpose	B rejects Video Share invitation.
Pre-conditions Scenario	1. A and B are in an active voice call, 2. Capabilities exchange has taken place confirming Video Share is possible. 3. Both devices have front or rear facing camera.
Test procedure	1. B rejects A’s invite to live Video Share. It is not possible for A to start sharing. 2. A initiates a capability exchange to verify B’s status.
Expected results **Post-conditions**	The call remains without Video Share.

Deep inspection	1. Verify signalling flow (invite + rejection). 2. Verify options exchange following cancelation (capability update).

Test case ID	**RCS_8_1_6_2**
Related test cases	
Feature	CONTENT SHARING - Video Share
Purpose	A terminates voice call during ongoing Video Share.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. A is currently sharing a video with B (currently transferring).
Test procedure	A ends the call while sharing.
Expected results **Post-conditions**	The Video Share is terminated (video no longer displayed).
Deep inspection	1. Verify share session establishment (including SDP negotiation). 2. Verify graceful termination of the share session (signalling) via traces. 3. Verify graceful termination of the share session (media) via traces. 4. If a video session, verify H.264 support in SDP negotiation. 5. Verify RTCP always used. 6. Verify that dummy packets were sent.

Test case ID	**RCS_8_1_6_3**
Related test cases	
Feature	CONTENT SHARING - Video Share
Purpose	A cancels Video Share invitation before B accepts.

Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place, confirming Video Share is possible. 3. Both devices have front or rear facing camera.
Test procedure	1. A sends Video Share invitation to B but then decides to cancel it. 2. B gets notification that the Video Share invite has been cancelled. 3. B verifies capabilities.
Expected results **Post-conditions**	The call remains without Video Share.
Deep inspection	1. Verify invitation process. 2. Verify cancelation response.

Test case ID	**RCS_8_1_6_4**
Related test cases	
Feature	CONTENT SHARING - Video Share
Purpose	Graceful termination of Video Share session because of changing coverage.
Pre-conditions Scenario	1. A and B are in 3G coverage and in an active voice call. 2. A Video Share session is already in place from A to B.
Test procedure	B goes to 2G coverage; consequently, the Video Share session cannot be maintained. A stops sending video to B.
Expected results **Post-conditions**	A stops receiving the video from B.

Deep inspection	1. Verify share session establishment (including SDP negotiation). 2. Verify graceful termination of the share session (signalling) via traces. 3. Verify graceful termination of the share session (media) via traces. 4. If a video session, verify H.264 support in SDP negotiation.

Test case ID	**RCS_8_1_6_5**
Related test cases	
Feature	CONTENT SHARING - Video Share
Purpose	Non-graceful termination – receiver.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. There is an active Video Share session from A to B.
Test procedure	1. B goes out of coverage; therefore, the video sent from A fails to reach B. 2. A performs a capability exchange to verify the new status.
Expected results **Post-conditions**	The Video Share is terminated on both ends, and the Video Share capability is shown as unavailable.
Deep inspection	1. Verify graceful establishment of the Video Share session (signalling) via traces (active session) and cancellation of the other invitation. 2. Verify graceful establishment of the Video Share session (media) via traces (active session). 3. Verify H.264 support in SDP negotiation. 4. Verify RTP RR/timeout process. 5. Verify signalling/media termination. 6. Verify capability exchange following the event.

Test case ID	**RCS_8_1_6_6**
Related test cases	
Feature	CONTENT SHARING - Video Share
Purpose	Non-graceful termination – sender

Pre-conditions Scenario	1. A and B are in an active voice call. 2. There is an active Video Share session from A to B.
Test procedure	1. A goes out of coverage; therefore, B stops receiving the video. 2. B performs a capability exchange to verify the new status.
Expected results **Post-conditions**	The Video Share is terminated on both ends and the Video Share capability is shown as unavailable.
Deep inspection	1. Verify graceful establishment of the Video Share session (signalling) via traces (active session) and cancellation of the other invitation. 2. Verify graceful establishment of the Video Share session (media) via traces (active session). 3. Verify H.264 support in SDP negotiation 4. Verify RTP RR/timeout process. 5. Verify signalling/media termination. 6. Verify capability exchange following the event.

8.2 Image Share

8.2.1 Basic Image Share

Test case ID	**RCS_8_2_1_1**
Related test cases	
Feature	CONTENT SHARING - Image Share
Purpose	Image Share, two simultaneous sessions
Pre-conditions Scenario	1. A and B are in a voice call. 2. Capabilities exchange has taken place confirming Image Share is possible.
Test procedure	1. A invites B to receive a stored picture. 2. B accepts. 3. B invites A to receive a stored picture. 4. A accepts.

Expected results **Post-conditions**	1. Updated capabilities for both users. 2. Picture sharing starts. B can see picture sent by A. 3. Picture sharing starts. A can see picture sent by B.
Deep inspection	For each session: 1. Verify share session establishment (including SDP negotiation). 2. Verify graceful termination of the share session (signalling) via traces 3. Verify graceful termination of the share session (media) via traces.

Test case ID	**RCS_8_2_1_2**
Related test cases	
Feature	CONTENT SHARING - Image Share
Purpose	Two session invitations at the same time.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Image Share is possible. 3. Both devices have front or rear facing camera.
Test procedure	1. A invites B to receive a stored picture. Before B accepts B invites A to receive another stored picture. 2. B accepts. 3. A accepts.
Expected results **Post-conditions**	1. Updated capabilities for both users. 2. Picture sharing starts. B can see picture sent by A. 3. Picture sharing starts. A can see picture sent by B.
Deep inspection	For each session: 1. Verify share session establishment (including SDP negotiation). 2. Verify graceful termination of the share session (signalling) via traces. 3. Verify graceful termination of the share session (media) via traces.

Test case ID	**RCS_8_2_1_3**
Related test cases	

Feature	CONTENT SHARING - Image Share
Purpose	Two simultaneous sessions. A does NOT accept.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Image Share is possible.
Test procedure	1. A invites B to receive a stored picture. 2. B accepts. 3. B invites A to receive a stored picture. 4. A does NOT accept.
Expected results **Post-conditions**	1. Updated capabilities for both users. 2. Picture sharing starts. B can see picture sent by A. 3. B receives an error message, as A rejects the invitation.
Deep inspection	For successful session: 1. Verify share session establishment (including SDP negotiation). 2. Verify graceful termination of the share session (signalling) via traces. 3. Verify graceful termination of the share session (media) via traces. For non-successful session: 1. Verify invitation. 2. Verify cancellation process.

8.2.2 Multiparty call and Image Share

Test case ID	**RCS_8_2_2_1**
Related test cases	
Feature	CONTENT SHARING - Image Share
Purpose	A, B, and C are in a CS call using Image Share (transfer not completed).

Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Image Share is possible.
Test procedure	1. A invites B to share a stored picture. 2. B accepts picture sharing. 3. A establishes a multiparty call with B and C.
Expected results **Post-conditions**	1. Voice call established between A and B. 2. Picture sharing starts. 3. A multiparty call is established among A, B, and C. 4. Picture sharing between A and B is completed. 5. RCS capabilities are updated for A.
Deep inspection	1. Verify graceful establishment of the Image Share session (signalling) via traces (active session). 2. Verify graceful establishment of the Image Share session (media) via traces (active session). 3. Verify H.264 support in SDP negotiation. 4. Verify session termination (sip/media). 5. Verify capability exchange following multiparty call establishment.

8.2.3 Call on hold and Image Share

Test case ID	**RCS_8_2_3_1**
Related test cases	
Feature	CONTENT SHARING - Image Share
Purpose	Call termination after transfer completed.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. A is currently sharing an image with B (the image had been already transferred and is still being displayed).
Test procedure	A ends the call.

Expected results **Post-conditions**	The Image Share is terminated (image no longer displayed).
Deep inspection	1. Verify share session establishment (including SDP negotiation). 2. Verify graceful termination of the share session (signalling) via traces. 3. Verify graceful termination of the share session (media) via traces. 4. If a video session, verify H.264 support in SDP negotiation.

8.2.4 Multi-device handling

Test case ID	**RCS_8_2_4_1**
Related test cases	
Feature	CONTENT SHARING - Image Share
Purpose	A shares image with B; B is using two devices.
Pre-conditions **Scenario**	1. B has two devices, B1 and B2, and both online and active. 2. A and B are in an active voice call; B is using his B1 device. 3. Capabilities exchange has taken place confirming Image Share is possible.
Test procedure	1. After the voice call is established, A starts Image Share, using stored image, to B. 2. B accepts the invite using his B2 device. 3. A transfers the stored image to B.
Expected results **Post-conditions**	1. B receives the image at his B2 device. 2. No more sharing capabilities available for A.
Deep inspection	1. Verify share session establishment (including INVITE forking and SDP negotiation). 2. Verify graceful termination of the share session (signalling) via traces. 3. Verify graceful termination of the share session (media) via traces.

8.2.5 Exception conditions

Test case ID	**RCS_8_2_5_1**
Related test cases	
Feature	CONTENT SHARING - Image Share
Purpose	B rejects Image Share session offer.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Image Share is possible. 3. Both devices have front or rear facing camera.
Test procedure	1. A invites B to receive a stored picture. 2. B does NOT accept session offer.
Expected results **Post-conditions**	1. Updated capabilities for both users, if applicable. 2. Picture sharing does not start. 3. A receives a message indicating "No answer" or similar after answer timeout expires.
Deep inspection	1. Verify invitation process. 2. Verify invitation timeout.

Test case ID	**RCS_8_2_5_2**
Related test cases	
Feature	CONTENT SHARING - Image Share
Purpose	A terminates voice call after image transfer completed.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. A is currently sharing an image with B (the image had been already transferred and is still being displayed).

Test procedure	A ends the call.
Expected results **Post-conditions**	The Image Share is terminated (image no longer displayed).
Deep inspection	1. Verify share session establishment (including SDP negotiation). 2. Verify graceful termination of the share session (signalling) via traces. 3. Verify graceful termination of the share session (media) via traces. 4. If a video session, verify H.264 support in SDP negotiation.

Test case ID	**RCS_8_2_5_3**
Related test cases	
Feature	CONTENT SHARING - Image Share
Purpose	A terminates voice call during ongoing image transfer.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. A is currently sharing an image with B (currently transferring).
Test procedure	A ends the call while sharing.
Expected results **Post-conditions**	The Image Share is terminated (transfer cancelled).
Deep inspection	1. Verify share session establishment (including SDP negotiation). 2. Verify graceful termination of the share session (signalling) via traces. 3. Verify graceful termination of the share session (media) via traces. 4. If a video session, verify H.264 support in SDP negotiation.

Test case ID	**RCS_8_2_5_4**
Related test cases	
Feature	CONTENT SHARING - Image Share
Purpose	A cancels Image Share invitation before B accepts.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Image Share is possible.
Test procedure	1. A invites B to receive a stored picture. 2. A sends cancellation before B accepts.
Expected results **Post-conditions**	1. Updated capabilities for both users, if applicable. 2. Picture sharing does not start. 3. B receives cancellation notification.
Deep inspection	1. Verify invitation process. 2. Verify invitation cancelation.

Test case ID	**RCS_8_2_5_5**
Related test cases	
Feature	CONTENT SHARING - Image Share
Purpose	A cancels Image Share while image transfer is in progress.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Image Share is possible.

Test procedure	1. A invites B to receive a stored picture. 2. B accepts invitation. 3. A sends cancellation while sharing.
Expected results **Post-conditions**	1. Updated capabilities for both users, if applicable. 2. Picture sharing starts. 3. Picture sharing ends. 4. B receives cancellation notification.
Deep inspection	1. Verify share session establishment (including SDP negotiation). 2. Verify MSRP/session cancellation and session settlement. 3. Verify options exchange following failure.

Test case ID	**RCS_8_2_5_6**
Related test cases	
Feature	CONTENT SHARING - Image Share
Purpose	Image Share cancelled; B is no longer available.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Image Share is possible.
Test procedure	1. A invites B to receive a picture. 2. B accepts the invitation. 3. B’s device is turned off (e.g., because of exhausted battery).
Expected results **Post-conditions**	1. Update capabilities for both users, if applicable. 2. Image sharing starts (transfer). 3. The transfer is interrupted; both users are notified.
Deep inspection	1. Verify share session establishment (including SDP negotiation). 2. Verify MSRP/session timeout and session settlement. 3. Verify options exchange following failure.

Test case ID	**RCS_8_2_5_7**
Related test cases	
Feature	CONTENT SHARING - Image Share
Purpose	Non graceful termination because of lack of coverage.
Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Image Share is possible.
Test procedure	1. A invites B to receive a stored picture. 2. B accepts. 3. A loses all kind of coverage during transfer.
Expected results **Post-conditions**	1. Updated capabilities for both users, if applicable. 2. Picture sharing starts. 3. Picture sharing ends.
Deep inspection	1. Verify share session establishment (including SDP negotiation). 2. Verify MSRP/session timeout and session settlement. 3. Verify options exchange following failure.

Test case ID	**RCS_8_2_5_8**
Related test cases	
Feature	CONTENT SHARING - Image Share
Purpose	Image Share size limit (receiver)
Pre-conditions Scenario	1. A and B are in an active voice call. 2. Capabilities exchange has taken place confirming Image Share is possible. 3. A is configured to have a larger file size transfer limit than B.

Test procedure	1. A selects to share an image with B. 2. A selects a file larger than B's maximum file size (part of the configuration). NOTE: To facilitate self-accreditation this test can be simulated.
Expected results **Post-conditions**	1. The invitation is sent across to B. 2. B receives a warning with the only option to reject the session. 3. The invitation is rejected.
Deep inspection	1. Verify that the invitation is sent. 2. Verify the invitation rejection and session settlement.

9 IP Voice and Video Call

9.1 IP Voice Call

9.1.1 Two-party IP Voice Call over NNI

Test Case ID	**RCS_9_1_1_1**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Voice Call
Purpose	Validate that the RCS client can originate an IP Voice call that is terminated as an IP Voice call.
Pre-conditions Scenario	A and B are in RCS-VoLTE or RCS-VoHSPA mode.
Test procedure	1. A initiates an IP Voice call with B. 2. B answers the IP Voice call from A. 3. A ends the IP Voice call with B.

Expected results **Post-conditions**	A two-party IP Voice call is successfully initiated, connected and ended between A and B.
Deep inspection	1. Verify the establishment of the IP Voice call between A and B via traces. Note: UE and networks follow 3GPP TS 24.229, 3GPP TS 24.173, and 3GPP TS 26.114 for establishment and termination of a call. 2. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel ○ Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel” and ○ Only audio media in SDP in 200 OK response • RTP in session established using SIP INVITE 3. Verify session terminations via traces.

Test case ID	**RCS_9_1_1_2**
Related Test Cases	**RCS_9_1_1_1**
Feature	IP VOICE AND VIDEO CALL - IP Voice Call
Purpose	Validate that the RCS client can originate an IP Voice call that is terminated as a CS call.
Pre-conditions Scenario	1. A is in RCS-VoLTE or RCS-VoHSPA mode. 2. B is in RCS-CS mode. 3. B’s network terminates the IP Voice call as a CS call.
Test procedure	1. A initiates an IP Voice call with B. 2. B answers a voice call from A. 3. A ends the voice call with B.
Expected results **Post-conditions**	A two-party IP Voice/CS voice call is successfully initiated, connected, and ended between A and B.
Deep inspection	1. Verify the establishment of the IP Voice call between A and B via traces. Note: UE and networks follow 3GPP TS 24.229, 3GPP TS 24.173, and 3GPP TS 26.114 for establishment and termination of a call. 2. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel

	○ Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel" and ○ Only audio media in SDP in 200 OK response 3. RTP in session established using SIP INVITE. 4. Verify session terminations via traces.

Test case ID	**RCS_9_1_1_3**
Related Test Cases	**RCS_9_1_1_1**
Feature	IP VOICE AND VIDEO CALL - IP Voice Call
Purpose	Validate that the RCS client can originate an IP Voice call that is terminated as an RCS IP Voice call.
Pre-conditions Scenario	1. A is in RCS-VoLTE or RCS-VoHSPA mode. 2. B is in RCS-CS or RCS-AA mode. Note: B's network terminates the IP Voice call as a RCS IP Voice call.
Test procedure	1. A initiates an IP Voice call with B. 2. B answers a voice call from A. 3. A ends the voice call with B.
Expected results **Post-conditions**	A two-party IP Voice/RCS IP Voice call is successfully initiated, connected, and ended between A and B.
Deep inspection	1. Verify the establishment of the IP Voice call between A and B via traces. Note: UE and networks follow 3GPP TS 24.229, 3GPP TS 24.173, and 3GPP TS 26.114 for establishment and termination of a call. 2. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel ○ Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel" and ○ Only audio media in SDP in 200 OK response 3. RTP in session established using SIP INVITE. 4. Verify session terminations via traces.

Test case ID	**RCS_9_1_1_4**
Related Test Cases	

Feature	IP VOICE AND VIDEO CALL - IP Voice Call
Purpose	Validate that the RCS client can place an IP Voice call on hold.
Pre-conditions Scenario	1. A and B are on an active two-party IP Voice call. 2. A and B are in RCS-VoLTE mode.
Test procedure	1. A places call on hold. 2. A resumes call. 3. B places call on hold. 4. B resumes call.
Expected results **Post-conditions**	A two-party IP Voice call is successfully placed on and off hold by both parties.
Deep inspection	Verify conformance with 3GPP TS 24.610.

Test case ID	**RCS_9_1_1_5**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Voice Call
Purpose	Validate that the RCS client can originate a two-party RCS IP Voice call.
Pre-conditions Scenario	A and B are in RCS-CS or RCS-AA mode.
Test procedure	1. A initiates a RCS IP voice call with B. 2. B accepts voice call from A. 3. A ends voice call with B.

Expected results **Post-conditions**	A two-party RCS IP voice call is successfully initiated, connected, and ended between A and B.
Deep inspection	1. Verify the establishment of the RCS IP voice call between A and B via traces. Note: UE and networks follow 3GPP TS 24.229, 3GPP TS 24.173, and 3GPP TS 26.114 for establishment and termination of a call. 2. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall ○ Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel" and ○ Only audio media in SDP in 200 OK response • RTP in session established using SIP INVITE 3. Verify session terminations via traces.

Test case ID	**RCS_9_1_1_6**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Voice Call
Purpose	Validate that the RCS client can place an active RCS IP Voice call on hold.
Pre-conditions Scenario	1. A and B are on an active two-party RCS IP Voice call. 2. A and B are in RCS-CS or RCS-AA mode.
Test procedure	1. A places call on hold. 2. A resumes call. 3. B places call on hold. 4. B resumes call.
Expected results **Post-conditions**	A two-party RCS IP Voice call is successfully placed on and off hold by both parties.
Deep inspection	Verify conformance with 3GPP TS 24.610.

Test case ID	**RCS_9_1_1_7**
Related Test Cases	**RCS_9_1_1_5**
Feature	IP VOICE AND VIDEO CALL - IP Voice Call
Purpose	Validate that the RCS client can originate an RCS IP Voice call that is terminated as a CS call.
Pre-conditions Scenario	1. A is in RCS-CS or RCS-AA mode. 2. B is in RCS-CS mode.
Test procedure	1. A initiates a RCS IP voice call with B. 2. B network terminates the call as a CS call to B. 3. B answers a voice call from A. 4. A ends the voice call with B.
Expected results **Post-conditions**	An RCS IP Voice call is successfully initiated, connected, and ended between A and B.
Deep inspection	1. Verify the establishment of the RCS IP Voice call between A and B via traces. Note: UE and networks follow 3GPP TS 24.229, 3GPP TS 24.173, and 3GPP TS 26.114 for establishment and termination of a call. 2. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall ○ Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel" and ○ Only audio media in SDP in 200 OK response 3. RTP in session established using SIP INVITE. 4. Verify session terminations via traces.

Test case ID	**RCS_9_1_1_8**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Voice Call
Purpose	Validate that the RCS client can originate a RCS IP Voice call that is terminated as a VoLTE call.

Pre-conditions Scenario	1. A is in RCS-CS or RCS-AA mode. 2. B is in RCS-VoLTE or RCS-VoHSPA mode. 3. B network terminates the call as a VoLTE call to B.
Test procedure	1. A initiates an RCS IP voice call with B. 2. B answers a voice call from A. 3. A ends the voice call with B.
Expected results **Post-conditions**	An RCS IP voice call is successfully initiated, connected, and ended between A and B.
Deep inspection	1. Verify the establishment of the RCS IP Voice call between A and B via traces. Note: UE and networks follow 3GPP TS 24.229, 3GPP TS 24.173, and 3GPP TS 26.114 for establishment and termination of a call. 2. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall ○ Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel" and ○ Only audio media in SDP in 200 OK response 3. RTP in session established using SIP INVITE. 4. Verify session terminations via traces.

Test case ID	**RCS_9_1_1_9**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Voice Call
Purpose	Validate that the RCS client can ignore an inbound IP Voice call.
Pre-conditions Scenario	A and B are in RCS-VoLTE or RCS-VoHSPA mode. Note: Voicemail service does not exist for B.
Test procedure	1. A initiates an IP Voice call with B. 2. B ignores inbound call from A (no answer).

Expected results **Post-conditions**	An ignored two-party IP voice call between A and B is cancelled.
Deep inspection	1. Verify IP Voice call between A and B via traces was not established (i.e. 408 or 487 error response occurs). 2. Verify the attempted establishment of the IP Voice call between A and B via traces. Note: UE and networks follow 3GPP TS 24.229, 3GPP TS 24.173, and 3GPP TS 26.114 for establishment and termination of a call. 3. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel ○ Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel” and ○ Only audio media in SDP in 200 OK response 4. RTP in session established using SIP INVITE. 5. Verify session termination between A and C via traces.

Test case ID	**RCS_9_1_1_10**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Voice Call
Purpose	Validate that the RCS client can ignore an inbound RCS IP Voice call.
Pre-conditions Scenario	1. A and B are visible and available on all devices. 2. A and B are in RCS-CS or RCS-AA mode. Note: Voicemail service does not exist for B.
Test procedure	1. A initiates an RCS IP Voice call with B. 2. B ignores the inbound RCS IP Voice call from A (no answer).
Expected results **Post-conditions**	An ignored two-party RCS IP voice call between A and B is cancelled.

Deep inspection	1. Verify that the IP Voice call between A and B via traces was not established (i.e., 408 or 487 error response occurs). 2. Verify the attempted establishment of the RCS IP Voice call between A and B via traces. Note: UE and networks follow 3GPP TS 24.229, 3GPP TS 24.173, and 3GPP TS 26.114 for establishment and termination of a call. 3. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall ○ Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel" and ○ only audio media in SDP in 200 OK response 4. RTP in session established using SIP INVITE.

Test case ID	**RCS_9_1_1_11**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Voice Call
Purpose	Validate that the RCS client can ignore an inbound RCS IP Voice call during a two-party RCS IP call.
Pre-conditions Scenario	1. A, B, and C are in RCS-CS or RCS-AA mode. 2. A and B are on an active RCS IP Voice call. Note: Voicemail service does not exist for A.
Test procedure	1. C initiates an RCS IP Voice with A. 2. A ignores the inbound call from C (i.e., no answer).
Expected results **Post-conditions**	A two-party RCS IP voice call between A and B is maintained and the call from C to A is cancelled.
Deep inspection	1. Verify that the IP Voice call between C and A via traces was not established (i.e. 408 or 487 error response occurs). 2. Verify the attempted establishment of the RCS IP Voice call between C and A via traces. Note: UE and networks follow 3GPP TS 24.229, 3GPP TS 24.173, and 3GPP TS 26.114 for establishment and termination of a call. 3. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall ○ Accept-Contact and Contact header fields containing the feature tag

	+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel” and o Only audio media in SDP in 200 OK response 4. RTP in session established using SIP INVITE.

Test Case ID	**RCS_9_1_1_12**
Related Test Cases	**RCS_9_1_1_1**
Feature	IP VOICE AND VIDEO CALL - IP Voice Call using WB-AMR codec at both ends
Purpose	Validate that the RCS client can originate an IP Voice call using WB-AMR codec that is terminated as an IP Voice call.
Pre-conditions Scenario	1. A and B are in RCS-VoLTE or RCS-VoHSPA mode. 2. UE of A is configured with WB-AMR codec. 3. UE of B is configured with WB-AMR codec 4. NNI RCS 5.1 Service Parameter ‘IP Voice Call Codec’ indicates ‘WB-AMR’.
Test procedure	1. A initiates an IP Voice call with B. 2. B answers the IP Voice call from A. 3. A ends the IP Voice call with B.
Expected results **Post-conditions**	A two-party IP Voice call using WB-AMR codec is successfully initiated, connected and ended between A and B.
Deep inspection	1. Verify the establishment of the IP Voice call between A and B via traces. Note: UE and networks follow [3GPP TS 24.229], [3GPP TS 24.173], and [3GPP TS 26.114] for establishment and termination of a call. 2. Verify: • SIP INVITE o P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel o Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel” and o Only audio media in SDP in 200 OK response • RTP in session established using SIP INVITE • Verify the support of WB AMR codec as part of SDP inspection. 3. Verify session terminations via traces.

Test Case ID	**RCS_9_1_1_13**
Related Test Cases	**RCS_9_1_1_1, RCS_9_1_1_12**
Feature	IP VOICE AND VIDEO CALL - IP Voice Call using NB-AMR codec at both ends
Purpose	Validate that the RCS client can originate an IP Voice call using NB-AMR codec that is terminated as an IP Voice call using NB-AMR codec.
Pre-conditions Scenario	1. A and B are in RCS-VoLTE or RCS-VoHSPA mode. 2. UE of A is configured with NB-AMR codec. 3. UE of B is configured with NB-AMR codec 4. NNI RCS 5.1 Service Parameter ‘IP Voice Call Codec’ indicates ‘NB-AMR’.
Test procedure	1. A initiates an IP Voice call with B. 2. B answers the IP Voice call from A. 3. A ends the IP Voice call with B.
Expected results **Post-conditions**	A two-party IP Voice call using NB-AMR codec is successfully initiated, connected and ended between A and B.
Deep inspection	1. Verify the establishment of the IP Voice call between A and B via traces. Note: UE and networks follow [3GPP TS 24.229], [3GPP TS 24.173], and [3GPP TS 26.114] for establishment and termination of a call. 2. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel ○ Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel” and ○ Only audio media in SDP in 200 OK response • RTP in session established using SIP INVITE • Verify the support of NB AMR codec as part of SDP inspection. 3. Verify session terminations via traces.

Test Case ID	**RCS_9_1_1_14**
Related Test Cases	**RCS_9_1_1_1, RCS_9_1_1_12**
Feature	IP VOICE AND VIDEO CALL - IP Voice Call with WB-AMR and NB-AMR codec at the originating end and NB-AMR codec at the terminating end
Purpose	Validate that the RCS client can originate an IP Voice call that is terminated as an IP Voice call using NB-AMR codec.

Pre-conditions Scenario	1. A and B are in RCS-VoLTE or RCS-VoHSPA mode. 2. UE of A is configured with WB-AMR and NB-AMR codecs. 3. UE of B is configured with NB-AMR codec. 4. NNI RCS 5.1 Service Parameter ‘IP Voice Call Codec’ indicates ‘WB-AMR’ and ‘NB-AMR’.
Test procedure	1. A initiates an IP Voice call with B. 2. B answers the IP Voice call from A. 3. A ends the IP Voice call with B.
Expected results **Post-conditions**	A two-party IP Voice call using NB-AMR codec is successfully initiated, connected and ended between A and B.
Deep inspection	1. Verify the establishment of the IP Voice call between A and B via traces. Note: UE and networks follow [3GPP TS 24.229], [3GPP TS 24.173], and [3GPP TS 26.114] for establishment and termination of a call. 2. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel ○ Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel” and ○ Only audio media in SDP in 200 OK response • RTP in session established using SIP INVITE • Verify the support of NB-AMR codec as part of SDP inspection. 3. Verify session terminations via traces.

Test Case ID	**RCS_9_1_1_15**
Related Test Cases	**RCS_9_1_1_1, RCS_9_1_1_12, RCS_9_1_1_14**
Feature	IP VOICE AND VIDEO CALL - IP Voice Call with NB-AMR codec at the originating end and WB-AMR codec and NB-AMR codec at the terminating end
Purpose	Validate that the RCS client can originate an IP Voice call that is terminated as an IP Voice call using NB-AMR codec.
Pre-conditions Scenario	1. A and B are in RCS-VoLTE or RCS-VoHSPA mode. 2. UE of A is configured with NB-AMR codec. 3. UE of B is configured with WB-AMR and NB-AMR codecs. 4. NNI RCS 5.1 Service Parameter ‘IP Voice Call Codec’ indicates ‘WB-AMR’ and ‘NB-AMR’.
Test procedure	1. A initiates an IP Voice call with B. 2. B answers the IP Voice call from A. 3. A ends the IP Voice call with B.

Expected results **Post-conditions**	A two-party IP Voice call using NB-AMR codec is successfully initiated, connected and ended between A and B.
Deep inspection	1. Verify the establishment of the IP Voice call between A and B via traces. Note: UE and networks follow [3GPP TS 24.229], [3GPP TS 24.173], and [3GPP TS 26.114] for establishment and termination of a call. 2. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel ○ Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel" and ○ Only audio media in SDP in 200 OK response • RTP in session established using SIP INVITE • Verify the support of NB-AMR codec as part of SDP inspection. 3. Verify session terminations via traces.

9.1.2 Multi-party RCS Voice Call over NNI

Test case ID	**RCS_9_1_2_1**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Voice Call
Purpose	Verify that an RCS client can make a multi-party IP Voice call.
Pre-conditions Scenario	1. A and B users are in RCS-VoLTE or RCS-VoHSPA mode. 2. C is an RCS user but not registered via IMS. Note: C's Network breaks out the IP Voice call to CS.
Test procedure	1. A initiates a multiparty IP Voice call with B and C. 2. A ends the IP Voice call with B and C.
Expected results **Post-conditions**	An IP Voice multi-party call is successfully established and ended among A, B, and C.

Deep inspection	1. Verify the establishment of the RCS IP Voice call session among A, B and C via traces. Note: UE and networks follow 3GPP TS 24.229, 3GPP TS 24.173, and 3GPP TS 26.114 for establishment and termination of a call. 2. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel ○ Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel" and ○ Only audio media in SDP in 200 OK response 3. RTP in session established using SIP INVITE. 4. Verify session terminations via traces.

Test case ID	**RCS_9_1_2_2**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Voice Call
Purpose	Verify that an RCS client can make a multi-party RCS IP Voice call.
Pre-conditions Scenario	1. A and B users are in RCS-CS or RCS-AA mode. 2. C is an RCS user but not registered via IMS. Note: C's Network breaks out the RCS IP Voice call to CS.
Test procedure	1. A initiates a multiparty RCS IP Voice call with B and C. 2. A ends the RCS IP Voice call with B and C.
Expected results **Post-conditions**	An RCS IP Voice multi-party call is successfully established and ended among A, B, and C.
Deep inspection	1. Verify the establishment of the RCS IP Voice call session among A, B, and C via traces. Note: UE and networks follow 3GPP TS 24.229, 3GPP TS 24.173, and 3GPP TS 26.114 for establishment and termination of a call. 2. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall ○ Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel" and ○ Only audio media in SDP in 200 OK response 3. RTP in session established using SIP INVITE.

	4. Verify session terminations via traces.

9.2 IP Video call

9.2.1 Two-party IP Video call over NNI

Test case ID	**RCS_9_2_1_1**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate that an RCS client can originate a two-party IP Video call.
Pre-conditions Scenario	1. A and B are IP Video call capable. 2. A and B are in RCS-VoLTE or RCS-VoHSPA mode. 3. Capabilities exchange has taken place confirming video calling is possible. 4. All devices have a front or rear facing camera.
Test procedure	1. A initiates an IP Video call with B. 2. B answers IP Video call from A. 3. Video call between A and B ends.
Expected results **Post-conditions**	A two-party IP Video call is initiated and successfully ends between A and B.
Deep inspection	1. Verify the establishment of the IP Video call via traces. 2. Verify SIP: • P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel • Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel” • Accept-Contact and Contact header fields containing the “video” capability indication and audio and video media in SDP in 200 OK response 3. Verify H.264 support in SDP negotiation (pre-condition). 4. If video call is with a front feed camera, verify lip synchronization complies with minimum performance criteria. 5. Verify session terminations via traces.

Test case ID	**RCS_9_2_1_2**
Related Test Cases	**RCS_9_2_1_1**
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate that the RCS client can place a two-party IP Video call on hold.
Pre-conditions Scenario	1. A and B are in an active two-party IP Video call. 2. A and B are in RCS-VoLTE or RCS-VoHSPA mode.
Test procedure	1. A places an IP Video call on hold. 2. A resumes the IP Video call. 3. B places the IP Video call on hold. 4. B resumes the IP Video call.
Expected results **Post-conditions**	A two-party IP Video call is successfully placed on and off hold by both parties.
Deep inspection	Verify conformance with 3GPP TS 24.610.

Test case ID	**RCS_9_2_1_3**
Related Test Cases	**RCS_9_2_1_1**
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate the RCS client can ignore an inbound IP Video call.
Pre-conditions Scenario	1. A and B are IP Video call capable. 2. A and B are in RCS-VoLTE or RCS-VoHSPA mode. 3. Capabilities exchange has taken place confirming video calling is possible. 4. All devices have a front or rear facing camera. 5. Voice / video mail service is inactive for B.

Test procedure	1. A initiates an IP Video call with B. 2. B ignores the inbound call from A (no answer).
Expected results **Post-conditions**	An ignored two-party IP Video call between A and B is cancelled.
Deep inspection	Verify that an IP Video call between A and B via traces was not established (i.e. 408 or 487 error response occurs).

Test case ID	**RCS_9_2_1_4**
Related Test Cases	**RCS_9_2_1_1**
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate the RCS client can accept only IP Voice from a inbound IP Video call.
Pre-conditions Scenario	1. A and B are IP Video call capable. 2. A and B are in RCS-VoLTE or RCS-VoHSPA mode. 3. Capabilities exchange has taken place confirming video calling is possible. 4. All devices have a front or rear facing camera.
Test procedure	1. A initiates an IP Video call with B. 2. B accepts the inbound IP Voice call from A.
Expected results **Post-conditions**	A rejected two-party IP Video call between A and B successfully connects as an IP Voice call.
Deep inspection	1. Verify the establishment of the IP Voice vs. IP Video call between A and B via traces. 2. Verify SIP INVITE: • P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel • Accept-Contact and Contact header fields containing the feature tag

	+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel" • Accept-Contact and Contact header fields containing the “video” capability indication and audio and video media in SDP in 200 OK response

Test case ID	**RCS_9_2_1_5**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate that a two-party IP Video call can gracefully degrade because of changing conditions.
Pre-conditions Scenario	1. There is an active IP Video call between A and B. 2. All devices have a front or rear facing camera.
Test procedure	B moves into 3G coverage; consequently, the two-party video session cannot be maintained.
Expected results **Post-conditions**	1. B stops sending video to A or receiving video from A. 2. The voice session downgrades to a CS call.
Deep inspection	1. Verify that the video portion of the IP Video call between A and B was removed via traces. 2. Verify graceful termination of the two-party video session and establishment of a CS call via traces.

Test case ID	**RCS_9_2_1_6**
Related Test Cases	**RCS_9_1_1_1**
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate that a two-party IP Video call can be downgraded to an IP Voice Call.

Pre-conditions Scenario	1. There is an active IP Video call between A and B. 2. All devices have a front or rear facing camera.
Test procedure	1. Video session is cancelled by A or B during an active IP Video call. 2. The IP Video call degrades to a IP Voice call between A and B.
Expected results **Post-conditions**	A two-party IP Video call degrades to a two-party IP Voice call between A and B.
Deep inspection	Verify that the video portion of the IP Video call between A and B was removed via traces.

Test case ID	**RCS_9_2_1_7**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate that the RCS client can upgrade an active two-party IP Voice call to a two-party IP Video call.
Pre-conditions Scenario	1. A and B are IP Video call capable. 2. A and B are in RCS-VoLTE or RCS-VoHSPA mode. 3. Capabilities exchange has taken place confirming video calling is possible. 4. All devices have a front or rear facing camera. 5. There is an active IP Voice call between A and B.
Test procedure	1. A upgrades from IP Voice call to an IP Video call with B. 2. B accepts the IP Video call from A.
Expected results **Post-conditions**	A two-party IP Voice call is upgraded to a two-party IP Video call between A and B.

Deep inspection	1. Verify SIP REINVITE: • P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel • Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel" • Accept-Contact and Contact header fields containing the "video" capability indication and audio and video media in SDP in 200 OK response 2. Verify H.264 support in SDP negotiation (pre-condition). 3. If Video call is with a front feed camera, verify that lip synchronization complies with minimum performance criteria.

Test case ID	**RCS_9_2_1_8**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate that the RCS client can originate a two-party RCS IP Video call.
Pre-conditions Scenario	1. A and B are RCS IP Video call capable. 2. A and B are in RCS-CS or RCS-AA mode. 3. Capabilities exchange has taken place confirming video calling is possible. 4. All devices have a front or rear facing camera.
Test procedure	1. A initiates an RCS IP Video call with B. 2. B answers the RCS IP Video call from A.
Expected results **Post-conditions**	A two-party RCS IP Video call is initiated between A and B.
Deep inspection	1. Verify establishment of the video session via traces. 2. Verify: SIP INVITE • P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall • Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel" • Accept-Contact and Contact header fields containing the "video" capability indication and audio and video media in SDP in 200 OK response

	• Separate RTP streams for audio and video in session established using SIP INVITE 3. Verify H.264 support in SDP negotiation (pre-condition) 4. If Video call is with a front feed camera, verify that lip synchronization complies with minimum performance criteria.

Test case ID	**RCS_9_2_1_9**
Related Test Cases	**RCS_9_2_1_1**
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate the RCS client can place a two-party RCS IP Video call on hold.
Pre-conditions Scenario	1. A and B are in an active two-party RCS IP Video call. 2. A and B are in RCS-CS or RCS-AA mode.
Test procedure	1. A places an RCS IP Video call on hold. 2. A resumes the RCS IP Video call. 3. B places the RCS IP Video call on hold. 4. B resumes the RCS IP Video call.
Expected results **Post-conditions**	A two-party RCS IP Video call is successfully placed on and off hold by both parties.
Deep inspection	Verify conformance with 3GPP TS 24.610.

Test case ID	**RCS_9_2_1_10**
Related Test Cases	**RCS_9_2_1_8**
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate that the RCS client can upgrade a two-party CS Voice call to a two-party RCS IP Video call, make before break.

Pre-conditions Scenario	1. A and B are RCS IP Video call capable. 2. A and B are in RCS-CS mode. 3. Capabilities exchange has taken place confirming voice and video calling is possible. 4. A and B are on an active two-party CS voice call. 5. All devices have a front or rear facing camera.
Test procedure	1. A initiates an RCS IP video call with B. 2. B accepts the RCS IP Video call from A.
Expected results **Post-conditions**	A two-party CS voice call is migrated to a two-party RCS IP Video call and successfully ends between A and B.
Deep inspection	1. Verify that the CS voice call between A and B ends after B accepts the RCS IP Video call. 2. Verify establishment of the IP Voice call between A and B via traces. Note: UE and networks follow 3GPP TS 24.229, 3GPP TS 24.173, and 3GPP TS 26.114 for establishment of a call. 3. Verify: SIP INVITE • P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall • Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel” • Accept-Contact and Contact header fields containing the “video” capability indication and audio and video media in SDP in 200 OK response • Separate RTP streams for audio and video in session established using SIP INVITE 4. Verify H.264 support in SDP negotiation (pre-condition). 5. If Video call is with a front feed camera, verify that lip synchronization complies with minimum performance criteria.

Test case ID	**RCS_9_2_1_11**
Related Test Cases	**RCS_9_2_1_8**
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate that the RCS client can upgrade a two-party CS Voice call to a two-party RCS IP Video call (break before make).
Pre-conditions Scenario	1. A and B are RCS IP Video call capable. 2. A and B are in RCS-CS mode.

	3. Capabilities exchange has taken place confirming voice and video calling is possible. 4. A and B are on an active two-party CS voice call. 5. All devices have a front or rear facing camera.
Test procedure	1. A initiates an RCS IP Video call with B. 2. B accepts the video call from A.
Expected results **Post-conditions**	A two-party CS voice call is migrated to a two-party RCS IP Video call and successfully ends between A and B.
Deep inspection	1. Verify that the CS voice call between A and B ends before B accepts RCS IP Video call. 2. Verify establishment of the IP Voice call between A and B via traces. Note: UE and networks follow 3GPP TS 24.229, 3GPP TS 24.173, and 3GPP TS 26.114 for establishment and termination of a call. 3. Verify: SIP INVITE • P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall • Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel” • Accept-Contact and Contact header fields containing the “video” capability indication and audio and video media in SDP in 200 OK response • Separate RTP streams for audio and video in session established using SIP INVITE 4. Verify H.264 support in SDP negotiation (pre-condition). 5. If Video call is with a front feed camera, verify that lip synchronization complies with minimum performance criteria.

Test case ID	**RCS_9_2_1_12**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate the RCS client can reject a two-party IP Video call upgrade request from a two-party IP Voice call
Pre-conditions Scenario	1. A and B are IP Video call capable. 2. A and B are in RCS-VoLTE or RCS-VoHSPA mode. 3. Capabilities exchange has taken place confirming video calling is possible. 4. All devices have a front or rear facing camera. 5. There is an active IP Voice call between A and B.

Test procedure	1. B attempts to upgrade the IP Voice call to an IP Video call with A. 2. A rejects the IP Video call upgrade from B. 3. The IP Voice call continues between A and B.
Expected results **Post-conditions**	A two-party IP Voice call is not upgraded to a two-party IP Video call and the IP Voice call continues between A and B.
Deep inspection	Verify that the IP Video call upgrade between A and B failed via traces (i.e., 488 error response occurs).

Test case ID	**RCS_9_2_1_13**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Video Muting.
Pre-conditions Scenario	1. A and B are in an active IP video call 2. A and B are in RCS-VoLTE mode.
Test procedure	1. A selects a video mute toggle via the application UI. 2. A suspends the video data transmission to B. 3. A selects the video mute toggle via the application UI. 4. A resumes the video data transmission to B.
Expected results **Post-conditions**	1. A stops sending video RTP packets to B. 2. A keeps sending RTCP Sender reports to B with zero bytes to indicate that A has muted the RTP stream(s). 3. A resumes sending video RTP packets to B. 4. A sends RTCP Sender reports to B with appropriate byte count to indicate that A has stopped muting the RTP stream(s).
Deep inspection	1. Verify that no video RTP packets are sent. 2. Verify that A keeps sending RTCP Sender reports to B with zero bytes for the muted the RTP stream(s). 3. Verify that video RTP packets are sent. 4. Verify that A keeps sending RTCP Sender reports to B with appropriate byte count for the video RTP stream(s).

Test case ID	**RCS_9_2_1_14**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate that the RCS client cannot make a two-party RCS IP Voice call to an RCS IP Video call only device; RCS IP Voice call fails.
Pre-conditions Scenario	1. A and B are RCS IP Video call capable. 2. B is an RCS IP Video Call only device; B is not RCS IP Voice call capable. 3. A and B are in RCS-CS or RCS-AA mode. 4. Capabilities exchange has taken place confirming voice and/or video calling is possible. 5. All devices have a front or rear facing camera.
Test procedure	1. A initiates an RCS IP Voice call with B. 2. B's phone does not ring. 3. RCS IP Voice call fails.
Expected results **Post-conditions**	A two-party RCS IP Voice call to an RCS IP Video Only device fails.
Deep inspection	Verify that an IP Voice call between A and B fails via traces (i.e. 488 error response occurs).

Test case ID	**RCS_9_2_1_15**
Related Test Cases	**RCS_9_1_1_7**
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate that the RCS client cannot make a two-party RCS IP Voice call to an RCS IP Video call only device; RCS IP Voice Call breaks out by the terminating network.
Pre-conditions Scenario	1. A and B are RCS IP Video call capable. 2. B is an RCS IP Video Call only device; B is not RCS IP Voice call capable. 3. A and B are in RCS-CS or RCS-AA mode. 4. Capabilities exchange has taken place confirming voice and/or video calling is possible. 5. All devices have a front or rear facing camera. 6. Terminating network breaks out RCS IP Voice calls to CS call to devices with a

	video only restriction.
Test procedure	1. A initiates an RCS IP Voice call with B. 2. B answers the CS call.
Expected results **Post-conditions**	An RCS IP Voice call is successfully terminated as a CS call.
Deep inspection	1. Verify the establishment of the RCS IP Voice session over the NNI via traces. Note: UE and networks follow 3GPP TS 24.229, 3GPP TS 24.173 and 3GPP TS 26.114 for establishment and termination of a call. 2. Verify SIP INVITE: • P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel.gsma.ipcall • Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel” and • only audio media in SDP in 200 OK response 3. RTP in session established using SIP INVITE.

Test case ID	**RCS_9_2_1_16**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate that call failure when a RCS client only accepts an IP Voice from an IP Video call only device.
Pre-conditions Scenario	1. A and B are RCS IP Video call capable. 2. A is an RCS IP Video Call only device; A is not RCS IP Voice call capable. 3. A and B are in RCS-CS or RCS-AA mode. 4. Capabilities exchange has taken place confirming voice and/or video calling is possible. 5. All devices have a front or rear facing camera.
Test procedure	1. A initiates an RCS IP Video call with B. 2. B’s phone rings. 3. B accepts the RCS IP Voice call. 4. The RCS IP Video call fails.

Expected results **Post-conditions**	A two-party RCS IP Video Call fails when video is not accepted by a non-video only restricted device.
Deep inspection	Verify that A's device/network terminates the call when B only accepts RCS IP Voice.

Test Case ID	**RCS_9_2_1_17**
Related Test Cases	**RCS_9_2_1_1**
Feature	IP VOICE AND VIDEO CALL - IP Video Call using H.264 CBP L1.2 codec at both ends
Purpose	Validate that the RCS client can establish an IP Video call using H.264 CBP L1.2 codec that is terminated as an IP Video call.
Pre-conditions Scenario	1. A and B are in RCS-VoLTE or RCS-VoHSPA mode. 2. UE of A is configured with H.264 CBP L1.2 codec. 3. UE of B is configured with H.264 CBP L1.2 codec. 4. NNI RCS 5.1 Service Parameter 'IP Voice Call Codec' indicates 'H.264 CBP L1.2'.
Test procedure	1. A initiates an IP Video call with B. 2. B answers the IP Video call from A. 3. A ends the IP Video call with B.
Expected results **Post-conditions**	A two-party IP Video call using H.264 CBP L1.2 codec is successfully initiated, connected and ended between A and B.
Deep inspection	1. Verify the establishment of the IP Voice call between A and B via traces. Note: UE and networks follow [3GPP TS 24.229], [3GPP TS 24.173], and [3GPP TS 26.114] for establishment and termination of a call. 2. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel ○ Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel" and "video" capability ○ Audio and video media in SDP in 200 OK response • RTP in session established using SIP INVITE

	• Verify the support of H.264 CBP L1.2 codec as part of SDP inspection. 3. Verify session terminations via traces.

Test Case ID	**RCS_9_2_1_18**
Related Test Cases	**RCS_9_2_1_1, RCS_9_1_1_17**
Feature	IP VOICE AND VIDEO CALL - IP Video Call using H.264 CBP L1.2 codec at one end only
Purpose	Validate that the RCS client cannot establish an IP Video call using H.264 CBP L1.2 codec if the terminating RCS client is not configured with support of H.264 CBP L1.2 codec.
Pre-conditions Scenario	1. A and B are in RCS-VoLTE or RCS-VoHSPA mode. 2. UE of A is configured with H.264 CBP L1.2 codec. 3. UE of B is not configured with H.264 CBP L1.2 codec. 4. NNI RCS 5.1 Service Parameter ‘IP Voice Call Codec’ indicates ‘H.264 CBP L1.2’.
Test procedure	1. A initiates an IP Video call with B. 2. B answers the IP Video call from A. 3. A ends the IP call with B.
Expected results **Post-conditions**	A two-party IP Video call using H.264 CBP L1.2 codec is not successfully initiated, connected and ended between A and B.
Deep inspection	1. Verify the establishment of the IP Voice call between A and B via traces of SDP negotiation. Note: UE and networks follow [3GPP TS 24.229], [3GPP TS 24.173], and [3GPP TS 26.114] for establishment and termination of a call. 2. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel ○ Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel” and “video” capability ○ Only audio media in SDP in 200 OK response • RTP in session established using SIP INVITE • Verify lack of H.264 CBP L1.2 codec support as part of SDP inspection. 3. Verify session terminations via traces.

Test Case ID	**RCS_9_2_1_19**
Related Test Cases	**RCS_9_2_1_1, RCS_9_1_1_17**
Feature	IP VOICE AND VIDEO CALL - IP Video Call established with H.264 CBP L1.2 codec where the terminating end supports a higher level codec than the originating end.
Purpose	Validate that an IP Video Call is established if an RCS client originates an IP Video call using H.264 CBP L1.2 codec and the terminating RCS client is configured to support a higher level H.264 CBP codec.
Pre-conditions Scenario	1. A and B are in RCS-VoLTE or RCS-VoHSPA mode. 2. UE of A is configured with H.264 CBP L1.2 codec. 3. UE of B is configured with a higher level H.264 CBP codec. 4. NNI RCS 5.1 Service Parameter ‘IP Voice Call Codec’ indicates ‘H.264 CBP L1.2’.
Test procedure	1. A initiates an IP Video call with B. 2. B answers the IP Video call from A. 3. A ends the IP Video call with B.
Expected results **Post-conditions**	A two-party IP Video call using H.264 CBP L1.2 codec is successfully initiated, connected and ended between A and B.
Deep inspection	1. Verify the establishment of the IP Video call between A and B via traces of SDP negotiation Note: UE and networks follow [3GPP TS 24.229], [3GPP TS 24.173], and [3GPP TS 26.114] for establishment and termination of a call. 2. Verify: • SIP INVITE ○ P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel ○ Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel” and “video” capability ○ Both video and audio media in SDP in 200 OK response • RTP in session established using SIP INVITE Verify H.264 CBP L1.2 codec support as part of SDP inspection: ***Example with A’s SDP Offer supporting level 1.2*** m=video 49268 RTP/AVP 99 a=rtpmap:99 H264/90000 a=fmtp:99 profile-level-id=**42e00c** ***Example with B’s SDP Answer supporting level 1.2 (B really supports a higher level H.264 CBP codec which implicitly means support for lower levels so no change to profile-level-id in SDP answer)*** m=video 49194 RTP/AVP 99

	a=rtpmap:99 H264/90000
	a=fmtp:99 profile-level-id=**42e00c**
	3. Verify session terminations via traces.

Test Case ID	**RCS_9_2_1_20**
Related Test Cases	**RCS_9_2_1_1, RCS_9_1_1_17**
Feature	IP VOICE AND VIDEO CALL - IP Video Call established with H.264 CBP L1.2 codec where the originating end supports a higher level codec than the terminating end
Purpose	Validate that an IP Video Call is established if an RCS client originates an IP Video call using a higher level H.264 CBP codec and the terminating RCS client is configured to support a H.264 CBP L1.2 codec and nothing higher.
Pre-conditions Scenario	1. A and B are in RCS-VoLTE or RCS-VoHSPA mode. 2. UE of A is configured with a higher level H.264 CBP codec than L1.2. 3. UE of B is configured with H.264 CBP L1.2 codec. 4. NNI RCS 5.1 Service Parameter ‘IP Voice Call Codec’ indicates support for at least H.264 CBP L1.2 codec but allows a higher level codec to be negotiated.
Test procedure	1. A initiates an IP Video call with B. 2. B answers the IP Video call from A. 3. A ends the IP Video call with B.
Expected results **Post-conditions**	A two-party IP Video call using H.264 CBP L1.2 codec is successfully initiated, connected and ended between A and B.
Deep inspection	1. Verify the establishment of the IP Video call between A and B via traces of SDP negotiation. Note: UE and networks follow [3GPP TS 24.229], [3GPP TS 24.173], and [3GPP TS 26.114] for establishment and termination of a call. 2. Verify: • SIP INVITE o P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel o Accept-Contact and Contact header fields containing the feature tag +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel” and “video” capability o Both video and audio media in SDP in 200 OK response • RTP in session established using SIP INVITE • Verify H.264 CBP L1.2 codec support as part of SDP inspection: ***Example with A’s Offer supporting a higher level H.264 CBP codec (L3.1 in this example)*** m=video 49268 RTP/AVP 99

	a=rtpmap:99 H264/90000
	a=fmtp:99 profile-level-id=**42e01f**
	Example with B's Answer supporting L1.2 (B negotiates down to L1.2)
	m=video 49194 RTP/AVP 99
	a=rtpmap:99 H264/90000
	a=fmtp:99 profile-level-id=**42e00c**
	3. Verify session terminations via traces.

9.2.2 Multi-party RCS Video call over NNI

Test case ID	**RCS_9_2_2_1**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate multi-party IP Video call capability
Pre-conditions Scenario	1. A and B are IP Video call capable. 2. A and B are in RCS-VoLTE mode. 3. Capabilities exchange has taken place confirming video calling is possible. 4. There is an active IP Video call between A and B. 5. All devices have a front or rear facing camera.
Test procedure	A initiates a multi-party IP Video call with B and C.
Expected results **Post-conditions**	Multi-party IP Video calls are successfully initiated among A, B, and C.
Deep inspection	1. Verify session set-up via traces. 2. Verify H.264 support in SDP negotiation (pre-condition). 3. If Video call is with a front feed camera, verify that lip synchronization complies with minimum performance criteria.

Test case ID	**RCS_9_2_2_2**
Related Test Cases	
Feature	IP VOICE AND VIDEO CALL - IP Video Call
Purpose	Validate multi-party RCS IP Video call capability
Pre-conditions Scenario	1. A and B are RCS IP Video call capable. 2. A and B are in RCS-CS or RCS-AA mode. 3. Capabilities exchange has taken place confirming video calling is possible. 4. There is an active RCS IP Video call between A and B. 5. All devices have a front or rear facing camera.
Test procedure	A initiates a multi-party RCS IP Video call with B and C.
Expected results **Post-conditions**	Multi-party RCS IP Video call are successfully initiated among A, B, and C.
Deep inspection	1. Verify session setup via traces. 2. Verify H.264 support in SDP negotiation (pre-condition). 3. If the Video call is with a front feed camera, verify that lip synchronization complies with minimum performance criteria.

10 Personal Network Blacklist

10.1 Standalone Message

10.1.1 Standalone message (Pager Mode) is screened out in terminating network

Test case ID	**RCS_10_1_1_1**
Related test case(s)	**RCS_7_1_1_1**
Feature	Personal Network Blacklist - Standalone Message
Purpose	Validate that standalone message (Pager Mode) is being screened out from the terminating network.

Pre-conditions **Scenario**	1. Both A and B are capable of standalone messaging. 2. The operators of both A and B have NNI interoperability of Personal Network Blacklist. 3. A is among the blacklisted users listed in 'rcs_pnb_standalone_blockedusers' of B in the Blacklist Policy Enforcement Function (BPEF).
Test procedure	A initiates sending of a standalone message in Pager Mode with 100 bytes of content towards B.
Expected results **Post-conditions**	1. The standalone message is not delivered to B because the BPEF of the terminating network determines that the delivery of the standalone message is to be blocked. 2. Verify no standalone message delivery and/or display notifications are returned to A if such delivery and/or display notifications are requested by A.
Deep inspection	1. Verify that the SIP MESSAGE request is sent from A network to B network and the P-Asserted-Identity header field does contain A's SIP URI. 2. Verify that a SIP 403 Forbidden response is received in the originating network with a warning header set to "122 Function not allowed".

10.1.2 Standalone message (Large Message Mode) is screened out in terminating network

Test case ID	**RCS_10_1_2_1**
Related test case(s)	**RCS 7_1_1_3**
Feature tested	Personal Network Blacklist - Standalone Message;
Purpose	Validate the SIP error code returned for standalone message (Large Message Mode) being screened out from terminating network.
Pre-conditions **Scenario**	1. Both A and B are capable of standalone messaging. 2. The operators of both A and B have NNI interoperability of Personal Network Blacklist. 3. A is among the blacklisted users list 'rcs_pnb_standalone_blockedusers' of B in the Blacklist Policy Enforcement Function (BPEF).
Test procedure	A initiates the sending of a standalone message in Large Message Mode, with more than 1500 bytes of content, towards B.

Expected results **Post-conditions**	1. The standalone message is not delivered to B as the BPEF of terminating network determines that the delivery of the standalone message is to be blocked. 2. Verify no standalone message delivery and/or display notifications will be returned to A if such delivery and/or display notifications are requested by A.
Deep inspection	1. Verify SIP INVITE request is sent from A network to B network and the P-Asserted-Identity header field does contain A's SIP URI. 2. Verify a SIP 403 Forbidden response is received in the originating network with a warning header set to "122 Function not allowed".

10.2 Chat session

10.2.1 Chat (1-to-1) invitation is screened out in terminating network

Test case ID	**RCS_10_2_1_1**
Related test case(s)	**RCS_7_2_1_1, RCS_7_2_2_2**
Feature	Personal Network Blacklist - Chat Session
Purpose	Validate that one-to-one Chat session invitation being screened out from the terminating network.
Pre-conditions **Scenario**	1. Both A and B are capable of chat service. 2. The operators of both A and B have NNI interoperability of Personal Network Blacklist. 3. A is among the blacklisted users listed in 'rcs_pnb_chat_blockedusers' of B in the Blacklist Policy Enforcement Function (BPEF).
Test procedure	A initiates one-to-one Chat session invitation towards B.
Expected results **Post-conditions**	1. The Chat invitation is not delivered to B as the BPEF of terminating network determines that the delivery of the Chat invitation is to be blocked. 2. Verify no Chat message delivery and/or display notifications will be returned to A if such delivery and/or display notifications are requested by A.
Deep inspection	1. Verify SIP INVITE request is sent from A network to B network and the P-Asserted-Identity header field does contain A's SIP URI. 2. Verify a SIP 403 Forbidden response is received in the originating network with a warning header set to "122 Function not allowed".

10.3 File transfer

10.3.1 File Transfer invitation is screened out in terminating network

Test case ID	**RCS_10_3_1_1**
Related test case(s)	**RCS_6_1_1_1**
Feature	Personal Network Blacklist - File Transfer
Purpose	Validate that MSRP-based File Transfer session invitation being screened out from the terminating network.
Pre-conditions Scenario	1. Both A and B are capable of MSRP-based file transfer service. 2. The operators of both A and B have NNI interoperability of Personal Network Blacklist. 3. A is among the blacklisted users listed in 'rcs_pnb_ft_blockedusers' of B in the Blacklist Policy Enforcement Function (BPEF).
Test procedure	A initiates MSRP-based File Transfer invitation towards B.
Expected results **Post-conditions**	The File Transfer invitation is not delivered to B as the BPEF of terminating network determines that the delivery of the File Transfer invitation is to be blocked. The File Transfer session is not established with B.
Deep inspection	1. Verify that the SIP INVITE request is sent from A network to B network and the P-Asserted-Identity header field does contain A's SIP URI. 2. Verify that a SIP 403 Forbidden response is received in the originating network with a warning header set to "122 Function not allowed".

Document Management

Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
0.1	May 9th 2013	Initial version	RCCTF #64	GSMA
1.0	Aug 6th 2013	Submitted to DAG#108 and PSMC#116 for approval	RCS & PSMC	GSMA

Other Information

Type	Description
Document Owner	RCC TF
Editor / Company	Vladimir Durovic 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.