RCS Release 1 Technical Realization

V2.0

14 February 2011

This is a non-binding permanent reference document of the GSM Association.

Security Classification – NON-CONFIDENTIAL GSMA Material

Table of Contents

1. INTRODUCTION AND SCOPE

1.1. Overview

This document describes the architecture and technical details needed for Rich Communication Suite, (RCS).

For a general overview of RCS including high-level requirements, please see the functional description document [FUNCDESC].

For configuration of RCS service, please see RCS management object specification [RCS MO]. RCS client implementations will comply with managed object specification.

1.2. In Scope

The scope of the document includes only the RCS Release 1. RCS Release 2 introduces new features related to supporting the Multi-Device Environment. For further information on how to make Release 1 clients working in the Multi-Device Environment with Release 2 clients and networks, see RCS Release 2 Technical Realization document.

The scope of this RCS release is defined in the [FUNCDESC] document and mainly encompasses:

1. Enhanced Address Book (EAB) including
   - Sharing of Social Presence Information with a list of contacts agreed by the end-user
   - Fetching capability of all contacts in the contact list
2. Content Sharing
   - Video Share
   - Image Share
3. File transfer
4. Enhanced Messaging

1.3. References

[24.229]	TS 24.229: Internet Protocol (IP) multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3, V7.13.1 http://www.3gpp.org
[IMENDORSE]	RCS Endorsement of OMA SIP/SIMPLE IM 1.0 https://infocentre.gsm.org/cgi-bin/grp_details.cgi?RD&group
[IMAGESHARE]	PRD IR.79 Image Share Interoperability Specification, 1.1, http://www.gsmworld.com/
[IR.65]	PRD IR.65 IMS Roaming & Interworking Guidelines, 3.6, http://www.gsmworld.com/
[OMADS]	DS Protocol, 1.2.1, http://www.openmobilealliance.org/
[PRESENCE]	Presence SIMPLE Specification, 1.1, http://www.openmobilealliance.org/

[Presence2.0_DDS]	Presence SIMPLE Data Specification, Draft Version 2.0, 15 October 2008 http://www.openmobilealliance.org/
[Presence2.0_TS]	Presence SIMPLE Specification, Draft Version 2.0, 14 October 2008 http://www.openmobilealliance.org/
[Presence_Content]	Presence Content XDM Specification, Draft Version 1.0, 01 October 2008 http://www.openmobilealliance.org/
[PRESENCEIG]	Implementation Guidelines for OMA Presence SIMPLE v1.1 Presence, http://www.openmobilealliance.org/
[PresenceXDM]	Presence XDM Specification, Approved Version 1.1 – 27 Jun 2008 http://www.openmobilealliance.org/
[RCS MO]	Management Object, Approved Version 1.2- February 25, 2010
[RFC 3261]	RFC 3261: SIP: Session Initiation Protocol, June 2002 http://www.ietf.org
[RFC 3966]	RFC 3966: The tel URI for Telephone Numbers, December 2004 http://www.ietf.org
[RFC 4482]	RFC 4482: CIPID: Contact Information for the Presence Information Data Format, July 2006 http://www.ietf.org
[RLSXDM]	Resource List Server (RLS) XDM Specification Approved Version 1.1 – 27 Jun 2008, http://www.openmobilealliance.org/
[SharedXDM]	Shared XDM Specification, Approved Version 1.1 – 27 Jun 2008 http://www.openmobilealliance.org/
[SIMPLEIM]	Instant Messaging using SIMPLE, 1.0, http://www.openmobilealliance.org/
[VIDEOSHARE]	PRD IR.74 Video Share Interoperability Specification, 1.3, http://www.gsmworld.com/
[XDM1.1_AD]	XML Document Management Architecture, Approved Version 1.1, 27 June 2008 http://www.openmobilealliance.org/
[XDM2.0_AD]	XML Document Management Architecture, Candidate Version 2.0, 16 September 2008 http://www.openmobilealliance.org/
[XDM1.1_Core]	XML Document Management (XDM) Specification, Approved Version 1.1, 27 June 2008 http://www.openmobilealliance.org/
[XDM2.0_Core]	XML Document Management (XDM) Specification, Candidate Version 2.0, 16 September 2008 http://www.openmobilealliance.org/
[XDMIG]	Implementation Guidelines for OMA XDM v1.1, http://www.openmobilealliance.org/

2. RCS ARCHITECTURE

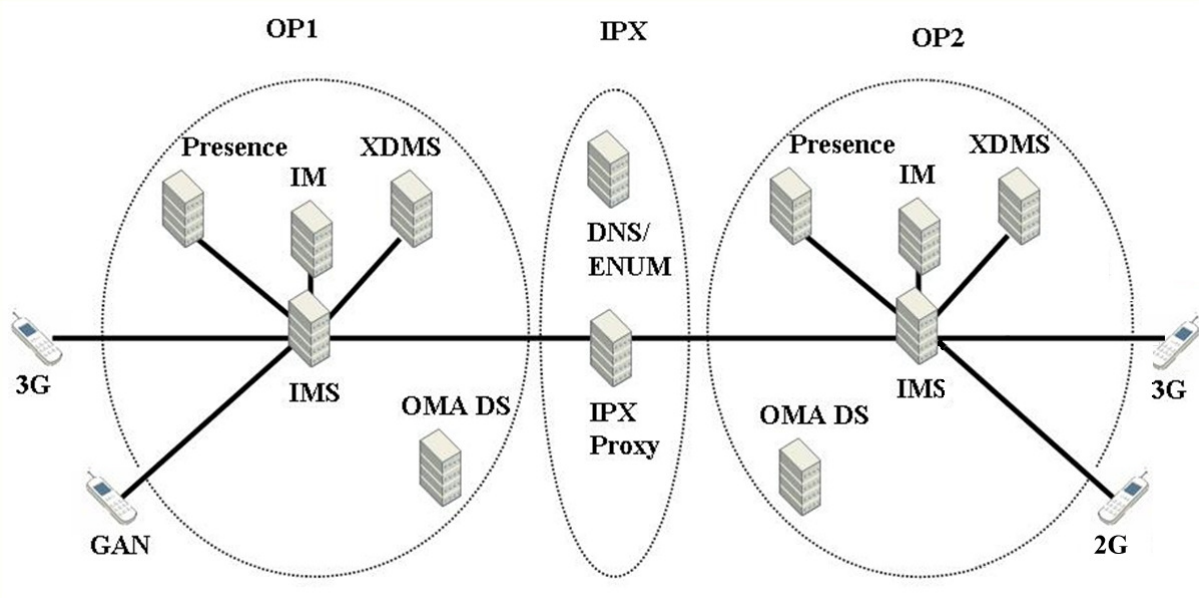


Figure 1: Simplified RCS Architecture

The figure above shows a simplified illustration of the overall architecture for RCS. Users can be connected to each other for the purpose of rich communication via IP based NNI using for example IPX (IP Exchange) handling the transport of both signalling and media between OP1 to OP2. Obviously RCS can be also offered as an intra-operator service even though this example shows an inter-operator scenario.

In order to assure interoperability, the RCS service is based on User-Network and Network-Network interfaces as defined in following specifications:

- Presence and Capability Discovery [PRESENCE]
- Video Share [VIDEOSHARE]
- Image Share [IMAGESHARE]
- Messaging [SIMPLEIM]

It should be noted that NNI is a part of the RCS concept. The usage of IPX, as shown in the Figure 1 is a way of NNI realisation, supporting the transport of protocols used in RCS NNI (as illustrated in Figure 2).

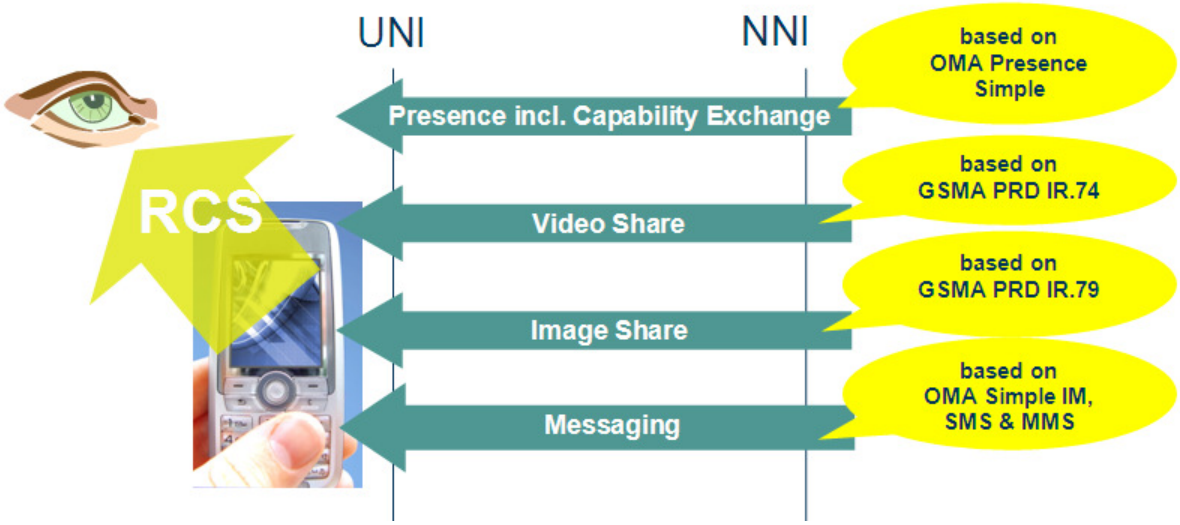


Figure 2: UNI's and NNI's of RCS

Note about Generic Access Network (GAN)

Generic Access to A/Gb interface provides a secure mechanism, using the SIM credentials, to access the mobile operator core network (both packet and circuit switched) using any unlicensed spectrum technology via a generic IP network. In fact, access to mobile operator core network via GAN is fully transparent from RCS perspective, and as such it does not lead to any particular limitation or impact from service point of view.

As a consequence, from access network perspective, this technology is fully part of the scope of RCS, whatever Release is addressed, irrespective of the release.

In this RCS Release, voice services are only supported on access networks offering natively the required CS voice capability (thus without the need for CS/PS voice conversion performed by MGW and so on components in the network). In practice, for example using Video Share and Image Share with non-mobile access networks is out of scope for this RCS release. GAN can be used for this RCS release, since it can be seen as an extension of the cellular network and does not require CS/PS voice conversion mechanisms.

3. GENERAL

3.1. Addressing

3.1.1. Overview

Telephone numbers in the legacy address book must be usable (regardless of whether RCS contacts have been enriched or not) for the identification of contacts of incoming and outgoing SIP requests. Note that these formats apply only to RCS services and should not impact the manner in which other services may have to convey phone numbers or even URIs. This also applies to the addresses used for Presence subscriptions (URI entries in XML documents), group chat communication (URI-lists, conference events and CPIM headers in MSRP) as well as for 1-1 communication.

3.1.2. Device Incoming SIP Request

From/P-Asserted-Identity:

For device incoming SIP requests, the address(es) of the contact are, depending on the type of request, provided as a URI in the body of the request (for example in case of a Watcher Info notification) or contained in the P-Asserted-Identity and/or the From headers. If P-Asserted-Identity is present, the From header will be ignored. The receiving client will try to extract the contact's phone number out of the following types of URI's:

- TEL URI's (for example tel:+1234578901 or tel:0234578901;phone-context=<*phonecontextvalue*>)
- SIP URI's with a "user=phone" parameter, the contact's phone number will be provided in the user part (for example. sip:+1234578901@operator.com;user=phone or sip:0234578901;phone-context=<*phonecontextvalue*>@operator.com;user=phone)

Note: SIP URI's without a "user" parameter where the user part is of the form "+digits" (for example sip:+1234578901@gsma.org) may also be supported if none of the previous types of URI are. However, this is not the recommended practice.

Note: Due to interworking with other (non-RCS) IMS based services, it may happen that still other types of URI's are received (for example alpha-numeric or anonymous URI's). How to deal with this is up to the client implementation.

3.1.3. Device Outgoing SIP Request

Identification of the contact:

The RCS client may use a telephone number (in local format for example 0234578901 or global format +1234578901) set in the RCS address book or a dial string entered by the user. This applies to the SIP Request-URI and the "To" header (as defined in [24.229]) for 1-1 communication, as well as to the URIs used in the recipient list included in outgoing SIP requests for group chat.

In case of international-format telephone number, the device should support tel-URI (for example tel:+12345678901) as defined in [RFC 3966] and SIP-URI (for example sip:+12345678901@domain;user=phone) with the user parameter set to "phone" as defined in [RFC 3261]. This must be configurable on the device based on [RCS MO] according to the operator's requirements or constraints related to national regulatory framework of SIP-SIP interconnection. If none of the above constraints apply, the use of tel URI is recommended since the domain name of the SIP-URI is not significant.

In case of non-international format telephone number, the RCS client should support tel-URI and SIP-URI (the user parameter should be set to "phone") with a phone-context value set as defined in [24.229] for home local numbers (for example tel:0234578901;phone-context=<home-domain-name>). Like the international number case, whether a TEL URI or a SIP URI is used should be configurable on the device according to the operator's requirements or constraints related to national regulatory framework of SIP-SIP interconnection. If none of the above constraints apply, the use of tel URI is recommended.

Self-Identification to the network and the addressed contact:

As telephone number is the main identifier to be used by the callee in order to identify the caller, the caller should set the P-Preferred-Identity header with a tel-URI which has been implicitly registered. The tel-URI could be provisioned in the device of the caller or it could be retrieved at the time of the registration using the P-Associated-URI header received from the network. This tel-URI contains the caller's telephone number in international format, for example, tel:+12345678901.

The From header should be set with the same URI as the P-Preferred-identity that is the tel-URI. This applies to the CPIM “From” header in MSRP SENDs used in group chat communication as well.

3.2. Registration

The RCS terminal shall register all feature tags as per service it supports, in the Contact header of a SIP REGISTER message. For information about the detailed structure of each feature tag related to RCS services, see reference [VIDEOSHARE], [IMAGESHARE] and [IMENDORSE]. For example, the following feature tags would be registered by an RCS terminal supporting Messaging, Video Share and Image Share:

- +g.oma.sip-im
- +g.3gpp.cs-voice
- +g.3gpp.iari-ref:urn:urn-xxx:3gpp-application.ims.iari.gsma-is

3.3. Session Response behaviour

When a RCS User declines a session invitation for example Chat, File Transfer, Video- and Image Share, the RCS client shall send a SIP 603 response to the INVITE request.

4. PRESENCE AND CAPABILITY DISCOVERY

4.1. Architecture

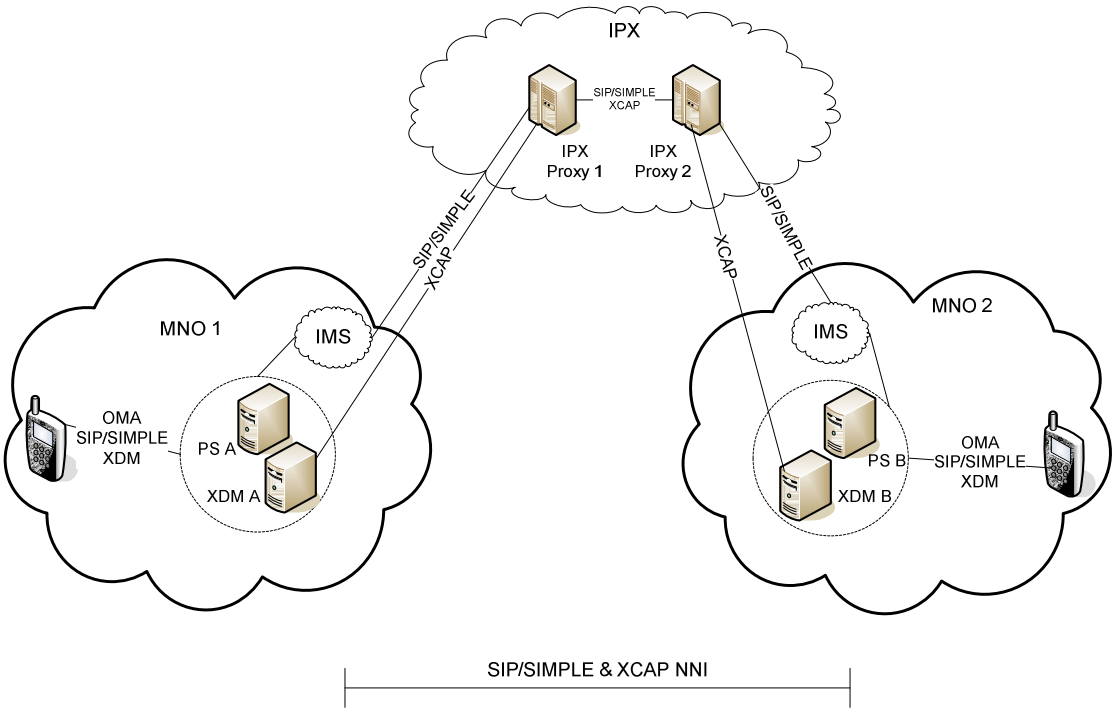


Figure 3: Overall Architecture of Presence as a part of RCS

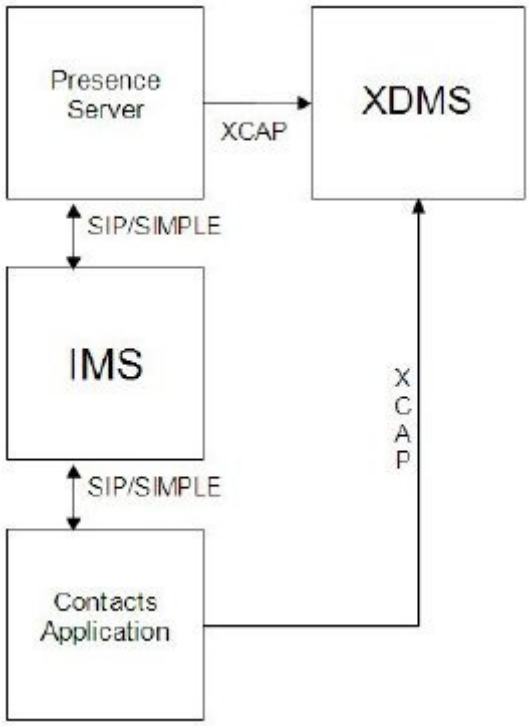


Figure 4: RCS Presence Architecture

Presence and capability architecture in RCS is based on [PRESENCE].

Users share their Social Presence Information (“Presence Enhanced Address Book”) Implemented using the Presence Simple protocol

Users share their communication capability information (“Capability Enhanced Address Book”) Implemented using the Presence Simple protocol

According to [IR.65], the interworking connection should be carried out via IMS core systems. There is therefore no requirement to interface Presence Servers directly.

Note: In the case of Image Share or Video Share, communication capabilities are discovered during a CS voice call based on SIP OPTIONS.

User’s contacts data is stored to network and retrieved from the network (“Network Address Book”) Implemented using the OMA Data Sync [OMADS]

Optimization of Presence & XDM enabler according to work ongoing for example in OMA PAG WG has to be taken into account as a very important design principle. It is also important to notice potential issues such as battery drain in the terminal caused by the general always-on functionality and the number of Presence & capability updates.

As a general rule, Shared XDMS as defined in [XDM1.1_AD] shall be used for storing all presence-related lists, for example the list of subscribed contacts (“buddy” list) and the presence authorization lists. In this way the RCS client need only operate on lists in Shared XDMS, and only initially set the documents in RLS XDMS and Presence XDMS.

4.2. Presence Data Model

4.2.1. Overview

Implementation guidelines for the size/length of Presence information elements given in [PRESENCEIG] should be followed.

The following chapters illustrate the details of the Presence Data Model.

4.2.2. Person

Attribute	Specification	Comment
Person: <presence> -> <person>	RFC 4479	According to the presence schema defined in the [OMA PRESENCE SIMPLE], person related information is modelled with the *person* element. Each client only publishes one person element.
Willingness: <person> -> <overriding-willingness> -> <basic>	OMA	The presentity terminal publishes this attribute in which it wants to indicate its willingness to communicate: “Open” = Willing “Closed” = Not Willing Attribute not present = Unknown
Icon: <person> -> <status-icon>	RFC 4480	It’s used as dynamic avatar. If the element is not present the client may chose to display icon stored in the address book.

		The picture shall not be included directly in the presence requests, but a HTTP URL shall be used. Presence Content XDMS procedures as specified in OMA Presence 2.0 and XDM 2.0 is used for uploading, publishing and retrieving the icon For further details see Chapter 4.8
Favourite Link: <person> -> <homepage>	RFC 4482	The <homepage> element provides a URI pointing to general information about the tuple or person, typically a web home page. This element can be used to convey one Favourite Link of the presentity.
Note: <person> -> <note>	RFC 4479	Presentity may write a piece of free text and/or to add emoticons to be shown to watchers in their contacts books The list of emoticons in RCS can be found in [IMENDORSE]
Timestamp: <person> -> <timestamp>	RFC 4479	Timestamp when the presence information was published.

Note: “Willingness” is sometimes indicated in a client as “Availability”. However since it is managed by the user himself and does not imply that communication is not possible, within OMA specifications this is considered as willingness. Availability indicates that on a technical level communication will be possible. Service Availability and Willingness are study items for later releases.

A service provider provisioning parameter is provided indicating whether or not the use of willingness is enabled by the operator. In case this parameter indicates that willingness is enabled, the RCS client will include in the presence document it publishes an OMA overriding-willingness element as specified in [Presence2.0_DDS] with the <basic> sub-element set to “Closed” if the user indicates that he’s not willing to communicate. Otherwise, the published presence document will indicate a value of “Open” for the <basic> sub-Element of overriding-willingness in case the provisioning parameter indicates that willingness is enabled. In case the parameter indicates that willingness is disabled, the RCS client will publish a presence document without any “<overriding-willingness>” element and ignore any “<overriding-willingness>” element it receives as a watcher.

4.2.3. Service

Attribute	Specification	Comment
Tuple: <presence> -> <tuple>	RFC 3863	According to the presence schema defined in the [OMA PRESENCE SIMPLE], services are presented with tuple elements.
Status <tuple> -> <status> -> <basic> -> Open	RFC 3863	Mandatory element in RFC 3863. Once a tuple element is published the value ‘open’ will always be used. It doesn’t have any particular meaning in RCS context.

Service-id <tuple> -> <service-description> -> <service-id>	OMA	*Service-description* element identifies a service and is described by a service-id and version. *Service-id* element contains a string that identifies a single service.
Version <tuple> -> <service-description> -> <version>	OMA	*Version* element contains the version number for the service, to identify different versions of the service (for example version number for specification number).
Contact <tuple> -> <contact>	RFC 3863	*Contact* element contains Presentity's communication address for the service. Contact address can be for example a TEL or SIP URI, depending on the service used. The use of the Contact element is optional (if used it has to be a global routable URI) since the client may use the URI stored in the Address Book when initiating communication with the presentity. RCS Presentities either do not insert any contact element or insert a contact element for which the address matches the one used for identifying itself in communication (see Chapter 3.1)
Timestamp <tuple> -> <timestamp>	RFC 3863	Timestamp when the presence information was published.

Service-descriptions for the Selected RCS Services

Registered Service-description values are listed in OMNA Presence <service-description> Registry:
http://www.openmobilealliance.org/Tech/omna/omna-prs-PidfSvcDesc-registry.aspx

CS Voice Call
Service-id: org.3gpp.cs-speech
Version: 1.0
Contact address type: TEL URI

CS Video Call
Service-id: org.3gpp.cs-videotelephony
Version: 1.0
Contact address type: TEL URI

Video Share
Service-id: org.gsma.videoshare
Version: 1.0
Contact address type: TEL / SIP URI

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

Session Mode Messaging
Service-id: org.openmobilealliance:IM-Session

Version : 1.0
Contact address type: TEL / SIP URI

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

Note: WCDMA access network is required for CS Video Call service.

4.2.4. Device

The Device part of presence is not a part of the RCS Release 1.

4.2.5. Example Document

Following is an example document created according to the rules specified in this document.

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
 xmlns:op="urn:oma:xml:prs:pidf:oma-pres"
 xmlns:opd="urn:oma:xml:pde:pidf:ext"
 xmlns:c="urn:ietf:params:xml:ns:pidf:cipid"
 xmlns:pdm="urn:ietf:params:xml:ns:pidf:data-model"
 xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid"
 entity="tel:+1234578901">

<tuple id="a0">
  <status><basic>open</basic></status>
  <op:service-description>
   <op:service-id>org.3gpp.cs-speech</op:service-id>
   <op:version>1.0</op:version>
  </op:service-description>
  <contact>tel: +1234578901</contact>
</tuple>

<tuple id="a1">
  <status><basic>open</basic></status>
  <op:service-description>
   <op:service-id>org.3gpp.cs-videotelephony</op:service-id>
   <op:version>1.0</op:version>
  </op:service-description>
  <contact>tel:+1234578901</contact>
</tuple>

<tuple id="a12">
  <status><basic>open</basic></status>
  <op:service-description>
   <op:service-id>org.gsma.videoshare</op:service-id>
   <op:version>1.0</op:version>
  </op:service-description>
  <contact>tel:+1234578901</contact>
</tuple>
```

```
<tuple id="a123">
  <status><basic>open</basic></status>
  <op:service-description>
   <op:service-id>org.openmobilealliance:IM-session</op:service-id>
   <op:version>1.0</op:version>
  </op:service-description>
  <contact>tel:+1234578901</contact>
</tuple>

<pdm:person id="a1233">
  <op:overriding-willingness>
    <op:basic>open</op:basic>
   </op:overriding-willingness>
   <rpid:status-icon opd:etag="26362">http://xcap.gsma.org/xcap-ap
service/org.openmobilealliance.pres-
content/users/sip:1234578901@gsma.org/oma_status-
icon/rcs_status_icon</rpid:status-icon>
   <c:homepage>http://example.com/~alice</c:homepage>
   <pdm:note>I'll be PAG</pdm:note>
</pdm:person>

</presence>
```

4.3. Rules to Ensure Backwards Compatibility

In order to maintain enough flexibility and not to impose potentially sub-optimal technical choices on future RCS releases, the presence parsing in an RCS Release 1 client should be sufficiently robust. Therefore the following guidelines should be taken into account in RCS presence parsing:

- Unknown or unsupported elements and tuples could be present in the document. In that case they should be ignored.
- Unknown Service-Id's could be present in the document. Tuples containing those should be ignored.
- Unknown service versions of known services could be present in the presence document. Tuples containing those should be ignored.
- The same service could occur multiple times in the presence document with different contact addresses. To cope with this case, the following behaviour shall be used for displaying and using the tuples:
    - If one of the tuples contains a contact address that corresponds to the presentity about which the presence document was received, all others shall be ignored.
    - Tuples that contain a contact (address) element which corresponds to another presentity (another contact in the contact-list of the user or another Tel-URI) shall be ignored.
    - Tuples containing contact elements with types of addresses that are not supported by the client for that service shall be ignored (for example messaging using an e-mail address while e-mail is not supported by the client)
    - If after applying the above rules, there are still multiple non-ignored tuples remaining for the service, all but the first shall be ignored.
    - If after applying the above rules there is a non-ignored tuple remaining behaviour shall be as follows

        - The capability to use the service for communication with the contact shall be announced to the user
        - If the remaining tuple contained no contact address or it matched the one of the presentity, the presentity's address will be used for setting up communication using that service
        - Otherwise the address contained in the contact element will be used for setting up the corresponding service
- When using RLS subscriptions, information could be contained on presentities that were not known to be part of the presence list (for example because the list was updated by another client or application). If the unexpected presentity is a known contact, it is advised that the client starts treating this contact as being presence enabled and tries to retrieve an updated presence list from the network.
- The Watcher shall follow the procedures defined in section 6.2 "Default Watcher Processing" of [Presence2.0_DDS]

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

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

4.4. Subscriptions and Authorization

4.4.1. Overview

When presence information is requested by a watcher of a presentity a SUBSCRIBE request is initiated (event package 'presence') according to [PRESENCE]. The watcher should be able to use the TEL URI to identify the presentity, see Chapter 3.1.

The support of RLS is mandatory for the clients and for the servers. Client shall conform to section 5.2.2.1 of the technical specification of [PRESENCE] and in addition to section 5.7.1 and 5.8 in [PRESENCEIG], section 5.1 in [XDMIG] and section 5.1.6 in [RLSXDM] as well. The XML documents shall follow the templates following later in this chapter.

Presence invitations are subject to reactive authorization to guarantee user privacy. This will allow the invited user (presentity) to accept, block or ignore an invitation to establish a presence relationship.

The presence authorization shall be symmetric, meaning that the inviting user also automatically authorizes the invited user to see his/her presence information, and that the invited user by accepting the presence invitation request both authorizes the inviting user to see his/her presence information and also subscribes to the inviting users presence information.

The RCS presentity shall be able to configure the presence authorization rules, which require the support in the RCS client and in the RCS Presence Server of [Presence-XDM]. The RCS client shall store a presence authorization document that follows [Presence-XDM] and the template rules described in section 5.8 in [PRESENCEIG].

In order for a presentity to be able to authorize the subscription of a watcher, the presentity needs to know which watcher(s) are trying to subscribe to the presence of the presentity. The RCS client and the Presence Server shall thus support section 5.3.1 and 5.4.4 of [PRESENCE].

When the subscription is authorized successfully, the presence server sends the presentity's presence document to the watcher by using the NOTIFY method as defined in [PRESENCE]. The format of the presence notification follows the Presence Data Model as describe above.

4.4.2. XML Document Structure

The Presence XDMS shall contain the following authorization rules, following the recommendations in [PRESENCEIG]:

- ”allow own” rule – allows subscriptions to own presence data
- ”confirm unlisted” rule – allows reactive authorization for contacts not yet allowed or blocked
- ”granted contacts” rule – contains those contacts that I'm subscribing to for presence (points to ”granted contacts” list in Shared XDMS)
- ”blocked contacts” – contains those contacts that I have blocked (points to ”blocked contacts” list in Shared XDMS)

The RLS XDMS shall for a RCS user contain a reference to the “rcs” list in Shared XDMS.

The Shared XDMS shall contain the following lists provided and managed by the RCS client:

- “*rcs*” list: This list includes all contacts you have a (symmetric) presence relation with. Commonly referred in RCS from both the “*buddylist*” and “*granted contacts*” lists as all your buddies shall be allowed to see your presence (symmetric).
- “*oma_buddylist*” list: Contains a reference to the “*rcs*” list where the actual buddies are stored. The “*oma_buddylist*” is not explicitly used and exists for compliance to [PRESENCEIG] and future extensibility.
- “*oma_grantedcontacts*” list: This list includes all contacts you have authorized to see your presence information. Contains a reference to “*rcs*” list
- “*oma_blockedcontacts*” list: Contains a reference to the “*rcs_blockedcontacts*” list where the actual permanently blocked contacts are stored and to the “*rcs_revokedcontacts*” list with the revoked users that are temporarily being blocked.
- *“rcs_blockedcontacts”* list: Contains all permanently blocked contacts
- *“rcs_revokedcontacts”* list: Contains all revoked contacts that are currently being blocked.
  Note: The “*rcs_revokedcontacts*” list is not intended to be shown to the end user. It is managed automatically.

Note: "*oma_grantedcontacts*" list and "*oma_buddylist*" list have the same content. That is why it is possible to point to the same "*rcs*" list.

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

Template:

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

  <cr:rule id="wp_prs_allow_own">
    <cr:conditions>
      <cr:identity>
        <cr:one id="tel:+1234578901"/>
      </cr:identity>
    </cr:conditions>
    <cr:actions>
      <pr:sub-handling>allow</pr:sub-handling>
    </cr:actions>
    <cr:transformations>
      <pr:provide-services>
        <pr:all-services/>
      </pr:provide-services>
      <pr:provide-persons>
        <pr:all-persons/>
      </pr:provide-persons>
      <pr:provide-devices>
        <pr:all-devices/>
      </pr:provide-devices>
      <pr:provide-all-attributes/>
    </cr:transformations>
  </cr:rule>

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

<cr:rule id="wp_prs_unlisted">
    <cr:conditions>
      <ocp:other-identity/>
    </cr:conditions>
    <cr:actions>
      <pr:sub-handling>confirm</pr:sub-handling>
    </cr:actions>
 </cr:rule>

 <cr:rule id="wp_prs_grantedcontacts">
    <cr:conditions>
       <ocp:external-list>
         <ocp:entry anc="http://xcap.gsma.org/resource-
lists/users/sip:1234578901@gsma.org/index/~~/resource-
lists/list%5B@name=%22oma_grantedcontacts%22%5D"/>
       </ocp:external-list>
    </cr:conditions>
    <cr:actions>
      <pr:sub-handling>allow</pr:sub-handling>
    </cr:actions>
    <cr:transformations>
      <pr:provide-services>
        <pr:all-services/>
      </pr:provide-services>
      <pr:provide-persons>
        <pr:all-persons/>
      </pr:provide-persons>
      <pr:provide-devices>
        <pr:all-devices/>
```

```
      </pr:provide-devices>
      <pr:provide-all-attributes/>
    </cr:transformations>
  </cr:rule>

  <cr:rule id="wp_prs_blockedcontacts">
    <cr:conditions>
       <ocp:external-list>
         <ocp:entry anc="http://xcap.gsma.org/resource-
lists/users/sip:1234578901@gsma.org/index/~~/resource-
lists/list%5B@name=%22oma_blockedcontacts%22%5D"/>
       </ocp:external-list>
    </cr:conditions>
    <cr:actions>
      <pr:sub-handling>block</pr:sub-handling>
    </cr:actions>
  </cr:rule>
</cr:ruleset>
```

RLS XDMS:
AUID: *rls-services*
Document name: *index*

Template:

```
<?xml version="1.0" encoding="UTF-8"?>
<rls-services xmlns="urn:ietf:params:xml:ns:rls-services">

   <service uri="sip:1234578901@gsma.org;pres-list=rcs">
      <resource-list>http://xcap.gsma.com/services/resource-
lists/users/sip:1234578901@gsma.org/index/~~/resource-
lists/list%5B@name=%22rcs%22%5D</resource-list>
      <packages>
         <package>presence</package>
      </packages>
   </service>

</rls-services>
```

Shared XDMS:
AUID: *resource-lists*
Document name: *index*

Template:

```
<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
      xmlns:xd="urn:oma:xml:xdm:xcap-directory"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<!-- The list oma_buddylist contains references to any individual list used according
to OMA IG for presence subscriptions. -->
    <list name="oma_buddylist">
     <external anchor="http://xcap.gsma.org/resource-
lists/users/sip:1234578901@gsma.org/index/~~/resource-
lists/list%5B@name=%22rcs%22%5D"/>
    </list>

<!-- The list oma_grantedcontacts contains the list of all granted contacts -->
    <list name="oma_grantedcontacts">
     <external anchor="http://xcap.gsma.org/resource-
lists/users/sip:1234578901@gsma.org/index/~~/resource-
lists/list%5B@name=%22rcs%22%5D"/>
    </list>

<!-- The list oma_blockedcontacts contains the list of all blocked contacts.  -->
    <list name="oma_blockedcontacts">
      <external anchor="http://xcap.gsma.org/resource-
lists/users/sip:1234578901@gsma.org/index/~~/resource-
lists/list%5B@name=%22rcs_blockedcontacts%22%5D"/>
      <external anchor="http://xcap.gsma.org/resource-
lists/users/sip:1234578901@gsma.org/index/~~/resource-
lists/list%5B@name=%22rcs_revokedcontacts%22%5D"/>
    </list>

<!-- The list of buddies the owner wants be able to get presence information for -->
<list name="rcs">
     <display-name>My presence buddies</display-name>

    </list>
<!-- The list of blocked contacts -->
    <list name="rcs_blockedcontacts">
     <display-name>My blocked contacts</display-name>

    </list>
<!-- The list of revoked contacts -->
    <list name="rcs_revokedcontacts">
     <display-name>My revoked contacts</display-name>
      <entry uri="tel:+123456" xd:last-modified="2008-12-24T14:32:14Z"/>
    </list>
</resource-lists>
```

Note: the entry in the “rcs_revokedcontacts” list is for illustrative purposes only. It is included as an example since it deviates slightly from the standard list usage.

4.4.3. Client Procedures, Initiation of Presence Sharing

When initiating a presence sharing request, the inviting user’s RCS client adds the invited user’s URI to the “*rcs*” list in Shared XDMS according to the procedures in [Shared-XDM].

Note: When adding the invited user’s URI to the “*rcs*” list, the RCS client shall check if the URI is included in the “*rcs_blockedcontacts*” or “*rcs_revokedcontacts*” list. If so, the URI shall be removed from those lists.

When the invited user receives a notification to establish a presence relation, the user can either;

a) Accept the invitation, whereas the RCS client of the invited user adds the inviting User’s URI to the “*rcs*” list in Shared XDMS according to the procedures in [Shared-XDM]. An example of a signalling sequence for the accept case is available in Appendix B.1.
b) Block the invitation, whereas the RCS client of the invited user adds the inviting User’s URI to the “*rcs_blockedcontacts*“list in Shared XDMS according to the

procedures in [Shared-XDM]. An example of a signalling sequence for the block case is available in Appendix B.2.

c) Ignore the invitation, whereas the RCS client of the invited user removes the presence sharing invitation. An example of a signalling sequence for the ignore case is available in Appendix B.3.

d) Not answer the invitation. The presence sharing invitation is pending in the client until either “accepted”, “blocked” or “ignored”. In the signalling, there is no difference from the “ignore” case.

4.4.4. Client Procedures, Removal of Presence Sharing

When the user decides to end the presence relationship with one of his contacts, he has to use the revoke option in his device. This triggers a notification to the user as defined in [FUNCDESC] asking for confirmation. When this is indeed confirmed, the client will put the user on the “*rcs_revokedcontacts*” list, subsequently remove the user from the “rcs” list and remove the contact’s presence information from the cache as defined in Chapter 4.5. When putting an entry for the contact in the “*rcs_revokedcontacts*” list the client includes a last modified attribute that indicates the current time in UTC.

When a client notices it has been blocked by a contact with whom Social Presence was shared (that is the RLS notify indicates the subscription is in state “terminated” and the reason indicates “rejected”), it will remove the contact from the “rcs” list and remove the contact’s cached presence information.

All clients will process the “*rcs_revokedcontacts*” list periodically and remove those users that have been on it for long enough (for example several days). For that they will compare the last-modified attribute of the entries to the current time. Both the interval at which the list is checked and the period that a contact should remain in this list will be operator configurable client parameters.

With regards to the communication capabilities both clients should fall back to the procedures as defined in Chapter 4.9.

4.4.5. Authorizing XCAP Requests

XCAP requests need to be authorized by the XDMS. This authorization relies on an assertion of the identity of the requestor of an XCAP request.

The HTTP header (“X-XCAP-Asserted-Identity” and “X-3GPP-Asserted-Identity”) used to contain the asserted identity of a requestor of an XCAP request may depend on operational conditions (type of access used by the terminal, operator policy) for example different operators may apply different algorithms to assert the identity of a requestor of an XCAP request. Thus, for any Authorization check to be carried out by the XDMS, any of both “X-XCAP-Asserted-Identity” and “X-3GPP-Asserted-Identity” headers is accepted as a valid header containing the asserted identity of the requestor of the XCAP request inside the operator domain.
But in order to offer a unique inter-operator interface, the “X-3GPP-Asserted-Identity“is always conveyed between 2 operator domains, at the NNI interface.

When the terminal of a Watcher requests, via XDM/XCAP, some content (for example status-icon, refer to § 4.8) associated with the Presence document of a Presentity, the XDMS of the Presentity has to check whether the Watcher is authorized to access this content, according to the Presentity's Presence Subscription Rules.

As defined in section 4.4.2, the "rcs" list is granted this permission.

The “rcs” list can contain both SIP URI and Tel URI address of authorized watchers in an operator domain. To ensure both cases :at the NNI interface, the “X-3GPP-Asserted-Identity “ of the initiator of an XCAP request should contain both the sip URI and Tel URI of this user.

4.4.6. Conditional Event Notification

The support of conditional event notification is strongly recommended for the clients (i.e. Watcher and Watcher Information Subscriber) and for the servers (i.e. Presence Server and RLS) in order to optimize presence traffic at UNI and NNI.

A RCS Client should support subscription (SIP SUBSCRIBE) with conditional event notification, as defined in Chapter 5.2.6 and Chapter 5.3.2 of [Presence2.0_TS].

A RCS RLS should support subscription (SIP SUBSCRIBE) with conditional event notification, as defined in Chapter 5.2 of [Presence2.0_RLS_TS].

A RCS Presence Server should support notification (SIP NOTIFY) with conditional event notification, as defined in Chapters 5.5.3.8, 5.5.3.9 and 5.5.4.2 of [Presence2.0_TS].

A RCS RLS should support notification (SIP NOTIFY) with conditional event notification, as defined in Chapter 5.4 of [Presence2.0_RLS_TS].

4.5. Caching Presence Information

The caching of presence information is a client procedure.

The RCS client must be able to locally store the most up-to-date presence information (that have been received through notifications) of all of the user's contacts. This locally stored information must be handled as a persistent cache (that is the data shall not be erased when the terminal is switched-off).

Whenever the watcher receives an empty presence document for a contact in a SIP NOTIFY for an active Social Presence relation, the client shall use the cached presence information pertaining to this contact instead of this empty document.

4.6. Publish

Presence document is published by using the PUBLISH method as defined in [PRESENCE]. Format of the presence notification is as defined according to the Presence Data Model as describe above.

When the application is started on the terminal the client sends a PUBLISH request according to the Presence Data Model.

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

A presence modify request is sent using the ‘Sip-If-Match’ header according to [PRESENCE].

4.7. Storage of the SIP-Etag Value & Presence Source Device Switching On/Off

This sub-clause addresses proper device support mechanisms for the use of publications with long term expire value period (for example several days or even more).

The device hosting the Presence Source must be able to be powered off without cancelling active publication event state in the Presence Server. Hence this publication event state can continue being delivered to watchers as long as it has not expired.

For example, a watcher that was not connected for some period of time and that is switching on, will be notified with the latest presentity's Social Presence Information (if not expired in Presence Server), even if the presentity's device is not powered on at the particular time when the watcher is switching on.

The Presence Source must follow the recommendation given in section 5.2.2 ('Storage of SIP-Etag Value") of [PRESENCEIG] which addresses the storage of the value of the "SIP-Etag" header field in persistent storage space and the use of the stored value in conditional publications for the lifetime of the publication.

By this way, the Presentity can manipulate the previous publication event state after the device is powered on again, rather than creating a new publication event state in the Presence Server.

Note: If despite this procedure the document would expire before the RCS client comes online again, the client will issue a new publication including the locally cached values for the different presence elements including the Overriding-Willingness. Like for all other cases when it comes online, the client may first request the user whether he wants to update any of the values though.

4.8. Status Icon Handling

4.8.1. Presentity Side

The status icon shall be stored, updated, deleted and retrieved according to the OMA Presence and XDM 2.0 procedures. For the storage itself, the Presence Content XDMS as defined in [Presence_Content] shall be used including the application usage and document type that it introduces. RCS will only make use of the presence content XDMS for the storage of the status icon. Therefore the usage as defined in chapter 5.1.12.1 of [Presence_Content] is the only one that is applicable including all its associated restrictions. After storing, updating or deleting the icon, the presentity's client should publish an updated presence document. The updated presence document shall include the etag attribute in the status-icon element as described in [Presence2.0_DDS] in chapters 7.11.1.3 and 7.20.

The icon and the icon document shall have following characteristics:

Document Name	rcs_status_icon
Icon aspect_ratio (width:height)	3:4 or 4:3
Icon maximum dimensions	240x320
Icon minimum dimensions	60x80
Icon file type	gif (both static and animated), jpeg or png as defined in [Presence_Content]
Document maximum size	200kB

Note: Fixing the icon document name will ensure that for RCS usage only a single icon is stored in the network and thus that no unnecessary resources are required for the storage of multiple icons. Without this, the situation could occur that multiple icons are stored without possibility to manage them after a switch to a new client. Furthermore the fixing of the icon name will allow clients that are aware of the SIP URI of their contact to build the URI needed for the retrieval of the icon even if the contact is offline.

Note2: 200kB is not a mandatory size. It is only defined as a maximum. Smaller sizes are acceptable and can on average even be expected.

The other parameters are fixed in order to allow the client implementations to know what to expect.

4.8.2. Watcher Side

The link to the status icon that is received in the presence document of the contact will be processed as described in [Presence2.0_TS] chapter 5.2.5.3. When the etag attribute of the status-icon element doesn't match that of the cached icon, the client will download the updated icon. For that it will handle the link that it received in the presence document as defined in [XDM2.0_Core] chapter 6.1.1.1 and more specifically the 3rd paragraph: it will replace the XCAP root part of the link with the own XCAP root of the watcher. After downloading the icon, the RCS client shall cache it along with the etag in order to be able to process future notifies on the status of the contact as defined in [Presence2.0_TS] chapter 5.2.5.3.

4.8.3. Network Handling

Finally in the network the retrieval of the information referred to by the link to the status icon will be realized in an architecture as described in [XDM1.1_AD] with the addition of the Cross-Network Proxies and XDM-8 and NNI-1 interfaces defined in [XDM2.0_AD]. The required functionality of the Cross-Network Proxy is limited to the authorization, data transfer and routing of XCAP functionalities. The routing of search requests is not applicable to RCS. For RCS the supported protocols on the NNI-1 interface are limited to XCAP, "limited XQuery over HTTP" is not supported.

At the functionality level, this means that the identity provided by the Aggregation Proxy is not only shared on the XDM-4 and enabler specific reference points between the Aggregation Proxy and the Enabler specific XDMS as it is described in [XDM1.1_Core] chapter 6.4.1, but also on the XDM-8 and NNI-1 interfaces as it is described in [XDM2.0_Core] chapter 5.1.3. The Integrity and Confidentiality protection of [XDM1.1_Core] chapter 6.4.2 is extended to the NNI-1 interface as it is described in [XDM2.0_Core] chapter 5.1.4. Furthermore in addition to the functionality described in [XDM1.1_Core], the Aggregation Proxy shall route requests to the Cross-Network proxy as it is described in [XDM2.0_Core] chapter

6.3.1.1 and route the Cross-Network Proxy's responses back to the XDM client. The procedures for routing requests to the search proxy that are described in [XDM2.0_Core] chapter 6.3.1.1 are not applicable for RCS. Finally the functionality of the Cross-Network Proxy as it is described in [XDM2.0_Core] chapter 6.5 and subchapters shall be supported with the exception of all functionality related to the routing of Search Requests and Search Responses.

4.9. Service Capabilities

4.9.1. General Overview

The service capabilities are realized using the Presence Data Model described in Chapter 4.2 except for the capability indication during a call which is realized using the SIP OPTIONS mechanism (described in Chapter 4.9.5).

4.9.2. Publication of the Service Capabilities

The service capability information that are object of a SIP PUBLISH by the RCS client (service tuple) correspond to the services supported by the device but may be restricted by some operator settings on the UE (on for example the services that are allowed by the operator in the network).

4.9.3. Service Capabilities Retrieval

Except for the restriction described in Chapter 4.9.1, service capabilities of an RCS user can be retrieved by another RCS user via a presence subscription issued by his/her client, providing the pertaining Presence Authorization rules allow him to do so.

Thus, an RCS user is allowed to retrieve the service capabilities of contacts with whom he has established a Social Presence relationship.

RCS users may also retrieve the service capability information of contacts with whom they have not established a Social Presence relationship by means of anonymous fetch operations issued by their client (as described in section 7.1 of [PRESENCE]) which will result in a single NOTIFY request indicating the service capabilities of that contact. This information shall then be cached in the client. Anonymous fetch operation shall be supported in client, but it shall be configurable by the operator whether it is used or not (refer to section 4.9.4).

The trigger of a new fetch operation will depend on the client implementation, but it could for instance be done when the user opens the contact card or when a new contact is added. By receiving this indication of capabilities or an empty document, a client may highlight, subject to operator policies, a 'subscribe' icon in order to bring the end-users attention towards the fact that the remote user may be subject for presence subscription.

4.9.4. Operator Controlled Service Capabilities Handling

For an operator wanting or having to implement privacy restriction ("privacy sensitive network") the following items can be configured subject to operator policies:

1. Authorization for a Watcher to retrieve or not the service capabilities via an anonymous fetch. If unauthorized, the RCS rule for anonymous fetch of

Presence Information is not set as described in Chapter 4.4.2 but is set to provide an empty document (at least telling the fetching terminal that the contact address supports Presence) as shown below :

```
<! -- This rule allows no service capabilities to be sent for anonymous
requests -->
<! -- in order to realize the service capabilities control requirement -->
<! -- This rule replaces the default “wp_prs_block_anonymous” rule -->
<! -- Note: May be modified to only allow RCS specified services -->
  <cr:rule id="rcs_allow_services_anonymous">
    <cr:conditions>
      <ocp:anonymous-request/>
    </cr:conditions>
    <cr:actions>
      <pr:sub-handling>allow</pr:sub-handling>
    </cr:actions>
    <cr:transformations/>
  </cr:rule>
```

Thus, for such an operator, only buddies of users can get the "capabilities" information (together with the Social Presence Information). Setting such a rule may be done by the client or enforced by operator policy in the network

2. Authorization for the terminal of a user to use or not the fetch operation described in Chapter 4.9.3.
3. Authorization for the terminal to display or not to the end-user the information regarding the ability of other contacts to share Social Presence Information.

4.9.5. Service Capabilities Query during a Call

In addition to publishing service capabilities via Presence as described in Chapter 4.2, RCS supports the service capability query during a CS voice call. This mechanism is applied for [VIDEOSHARE] and [IMAGESHARE] services used together with CS voice call and the service capability indication is shown in the Call UI.

Capability query is performed according to the [VIDEOSHARE] and [IMAGESHARE] specifications. After the CS call is set up, the capabilities of the other terminal are queried to find out if the recipient is capable of supporting Video Share and/or Image Share session. This query is performed with the SIP OPTIONS method. A positive response to the query is sent using 200 OK and it contains also detailed information about supported media formats and so on. Both UEs perform this query. A negative response to the content sharing (VS/IS) query or invite shall be sent using SIP 480 (temporary unavailable) when no voice call exists in the queried or invited terminal, and when a voice call exists but not with the querying/inviting party.

4.9.6. UI Related Aspects for Image Share and Video Share

Icons can be used in the terminal UI to show the user that a Video Share (or Image Share) session towards this particular recipient can be set up, in case the recipient indicates support for the Video Share service (or Image Share respectively).

Note: The goal is to ensure that users should not be proposed in the terminal UI to initiate a Video Share (or Image Share) session when the result from the capability query is not positive. Hence, the user experience is preserved.

4.10. XDM Document Management

XDM documents can be updated without the involvement of the RCS client of this RCS release. Two types of changes are possible:

1. On the one hand it can happen that shared lists are updated by adding new entries, removing entries or updating entries.
2. On the other hand though, there is a more difficult category of changes that may be done by another client: structural changes to the documents (for example to support new options in the presence authorization).

In the former case, in order not to overwrite changes done for example by another client, either a conditional update should be done (per XCAP conditional operations as defined in [RFC4825] section 7.11) or the client should retrieve the latest status of the document before doing the update. A RCS client of this RCS release shall support one of these options when updating XDM documents.

To deal with the latter case (structural changes to a XDM document) which could occur when a RCS client of this RCS release is deployed in a future RCS environment (even though the future RCS releases should be backward compatible with previous releases), the RCS client shall go to a read-only mode with regards to all XDM documents when it detects such changes. Future RCS versions will indicate this by renaming the “rcs” shared list. If the list is not renamed, but structural changes were detected in documents in the presence and RLS XDMS, the RCS client will go to read-only mode only for the updated documents. In that case the RCS client indicates to the user that (s)he should use a client with an updated RCS version to carry out commands that require modifying any of such documents.

To cover for situations in which the user downgrades from a future RCS release to the use of an RCS client only (for example the end-user does not have a client with an updated RCS version or there is some blocked situation between the XDMC and XDMS), the RCS client shall offer to the user the possibility to remove all information stored in the XDMS’s and then create new documents based on its current status and RCS release. The removal of the documents shall be based on a retrieval of the complete list of documents using XCAP Directory requests and then removing all listed documents (thus including documents unknown to the RCS client of this RCS release) using relevant operation such as XCAP PUT/DELETE.

Should a device for its own internal use maintain a local copy of the Shared XDMS’s “resource-lists” document (see chapter 4.4.2) or the information contained therein, then it shall verify with the Shared XDMS whether its copy is still up to date in the following situations:

- When the client comes online
- When it receives a notification within the dialog of its RLS subscription indicating that the subscription to some contact is pending or even active and according to the locally maintained information, it isn’t aware that that user is part of the RCS buddy list.
  Note: this situation can occur, in case the user invited the contact to share social presence information from another client.
- When it receives a notification within the dialog of its watcher information subscription indicating that a subscription from a contact changed from the “pending” to the “active” or “terminated” state when no action was taken to authorize or block that subscription from the client. The state change to “Terminated” should of course only be taken into account for this case when the event triggering the state change indicates “rejected”.
  Note: this situation can occur when the user authorizes or blocks the

subscription from another client.

- When it receives a notification within the dialog of its RLS subscription indicating that the subscription to a contact that is presence enabled was terminated with reason “timeout” when no action was taken from the client to revoke the presence sharing with that contact.

Note: a device is not required to maintain a local copy of the Shared XDMS’s “resource-lists” document. In case it doesn’t do so, it can of course simply display the presence information it receives and it does not need to access the XDMS.

5. ADDRESS BOOK

5.1. Enhanced Address Book

Enhanced Address Book (EAB) is a contact book application that makes the contacts a network service and enriches the contacts data provided by the Network Address Book (NAB) by integrating presence information.

EAB uses the SIMPLE presence data model defined in IETF and referenced by OMA Presence Simple 1.1 specification to provide presence and service capability information.

The service capability information for the presentity is retrieved from the presentity’s presence document. The presence document contains a list of services that the presentity has.

How the RCS application (application’s user interface) shows this information is up to the application. However in order for the different RCS applications to interwork there must be a consensus of how the presence document is formed and interpreted. Thus, the presence schema defined in [PRESENCE] is used as a base for the RCS service capability as described in the previous chapter.

The EAB can include a search function for yellow/white pages, corporate directories and so on. This feature is based on normal browsing functionality utilizing a pre-configured URL for the search page plus vCard download.

5.2. Network Address Book

5.2.1. Overview

The RCS Network Address Book (NAB) provides a mechanism for users to store and manage their address book contacts in a network. Functionalities provided by NAB includes the following

Network Storage:
A network-based repository related to a user’s account Includes storing contacts belonging to User to a common repository

1. Synchronization:
Allow a user to have his/her local contacts/data objects in sync with NAB

2. Address book management (local copy):

- Create/Add: introducing a new contact/data object
- Modify : update of existing contact /content of data object or renaming
- Delete : removing contact from local

3. Back up/Restore

5.2.2. Functional Components

RCS NAB is based on [OMA DS] and can be described as consisting of the following functional components, see Figure 5

- A device that contains OMA Data Sync Client: use for sending/receiving Data synchronization requests/responses
- OMA DS Server that contains Data Sync Server : receives synchronization messages and also sends responses to the Data sync Client
- Network Repository : database/repository for the user's contacts in network
- NAB may also contain a notification entity that OMA DS Serve can use to trigger the device to initiate synchronization with the OMA DS Server. An example is shown in Figure 5

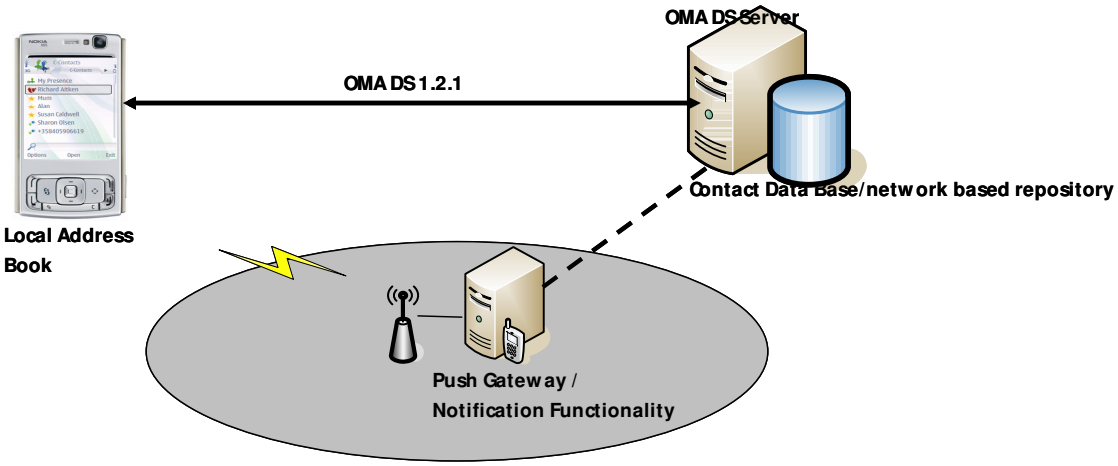


Figure 5: Illustrative Architecture for Network Address Book

For a User to be able to use the network address book services, the service provider that host the NAB service has to provisioned the NAB's address (that is a service specific URL is provided for this feature in the operator network) to the User's device.

5.2.3. Authentication

For an RCS user to access the NAB Service, [OMA DS] mandates support for both basic and MD5 authentication by the RCS user's device.

The MD-5 authentication involves following the MD-5 specification in [OMA DS] for server-based challenge to authentication

An example of basic authentication involves mobile network of service provider to append MSISDN, detected by mobile portal, as a HTTP header value to the HTTP/SyncML request sent to DS server for use in authenticating the requesting network entity. A secondary authentication credential, such as User-Agent string from HTTP/SyncML request can be used.

5.2.4. NAB Features

For the first synchronization between the device phone book and NAB or when the static device information has been updated, the device and the server SHALL exchange capabilities such as identifier, sync types supported, data store names, data format supported for each data store, and so on as defined in [OMA DS]

RCS user must be able to perform the following functions according to procedures defined in [OMA DS]. RCS user

- SHALL perform his/her first synchronization to the network address book by initiating Full Sync (Slow sync)
- SHALL be able to manage his/her address book (add, modify or delete contacts) on the local device by initiating Incremental sync (Two-way sync) to exchange information about any modification of the device database with the server's
- SHALL be able to backup local contacts to a network server, by initiating Refresh sync from client only (Backup sync)
- SHALL be able to restore local contact from a network backup server, by initiating Refresh sync from server only (Restore Sync)

Depending on RCS user or operator preference, synchronization/backup of local address book can be manual or schedule to be performed daily, weekly, monthly and so on.

Server-alerted sync is a function supported by RCS NAB, where the DS server alerts the client to perform a specific type of synchronization with the DS server. This sync type allows on-demand/on-event synchronization from networked services. For this functionality, RCS device must support the notification event package defined in [OMA DS].

RCS NAB shall support Suspend and Resume function as defined in [OMA DS] for the cases mentioned below:

- User initiated interruption/Pause
- Loss of network coverage or phone malfunction
- Notification

Whenever a DS server wants to trigger a device to initiate synchronization with the DS server, the DS server delivers a notification event package to a network Notification entity. How the DS server delivers this notification to the network notification entity is out of scope of this document.

The network notification entity is responsible for delivery the notification to the device(s). This is out of scope of this document

If a device receives notification for server-alerted sync, the device should initiate the requested synchronization even if the device has just completed synchronization with the DS server.

6. CONTENT SHARING

6.1. Video Share

Video Share shall follow [VIDEOSHARE].

It is strongly recommended that RCS Client supports video codec H264/AVC Baseline Profile Level 1b (using the RTP format in RFC 3984) and, when supported, lists this media format as first format being preferred in “m=” line in SDP (as defined in RFC 3264) for Video Share.

An RCS terminal might use the EAB in addition to sending OPTIONS to discover the other terminals Video Share capability before setting up the CS call.

The “already deployed terminals” option in [VIDEOSHARE] is not applicable in RCS.

6.2. Image Share

Image Share shall follow [IMAGESHARE]. Note that all RCS services using MSRP, including Image Share, shall align with MSRP usage as described in [IMENDORSE].

An RCS terminal might use the EAB in addition to sending OPTIONS to discover the other terminals Image Share capability before setting up the CS call.

Details for image format as specified in 3GPP TS 26.141 *IMS Messaging and Presence: Media Formats and codecs* will be followed.

7. FILE TRANSFER

File Transfer shall follow [SIMPLEIM] (using MSRP for file transport) as described in [IMENDORSE]. File Transfer is not linked to other services (for example CS-voice call) and can be used either during or outside of other communication sessions.

An RCS terminal might use the EAB to discover the other terminals File Transfer capability.

8. MESSAGING

8.1. Legacy Messaging

Legacy messaging is realized using existing SMS and MMS service. IM Pager Mode and IM Large Message Mode as of [SIMPLEIM] may be used for a future implementation of the SMS/MMS services for packet-only devices.

8.2. Chat

The RCS Chat service as defined in [FUNCDESC] is based on OMA SIMPLE IM Session mode. 1-to-1 chat shall use “One to One Session Mode messaging” and Group Chat the “Ad-Hoc Session Mode messaging” following [SIMPLEIM] as described in [IMENDORSE].

8.3. Other Messaging Functions

Conversational message view for messaging is an important feature of RCS. It is to be implemented in the terminal also for non-session based messaging types such as SMS and MMS.

Interworking between OMA SIMPLE IM based services and legacy services (SMS/MMS) is a study item for later Releases.

9. OTHER SERVICES

9.1. Multimedia CLI

MMCLI is a study item for later Releases.

9.2. Wideband Speech Communication Capabilities

It is recommended that RCS Client supports wideband speech communication capabilities based on Adaptive Multi Rate WideBand (AMR-WB) in CS voice call.

APPENDIX A: ITEMS FOR FURTHER STUDY (INFORMATIVE)

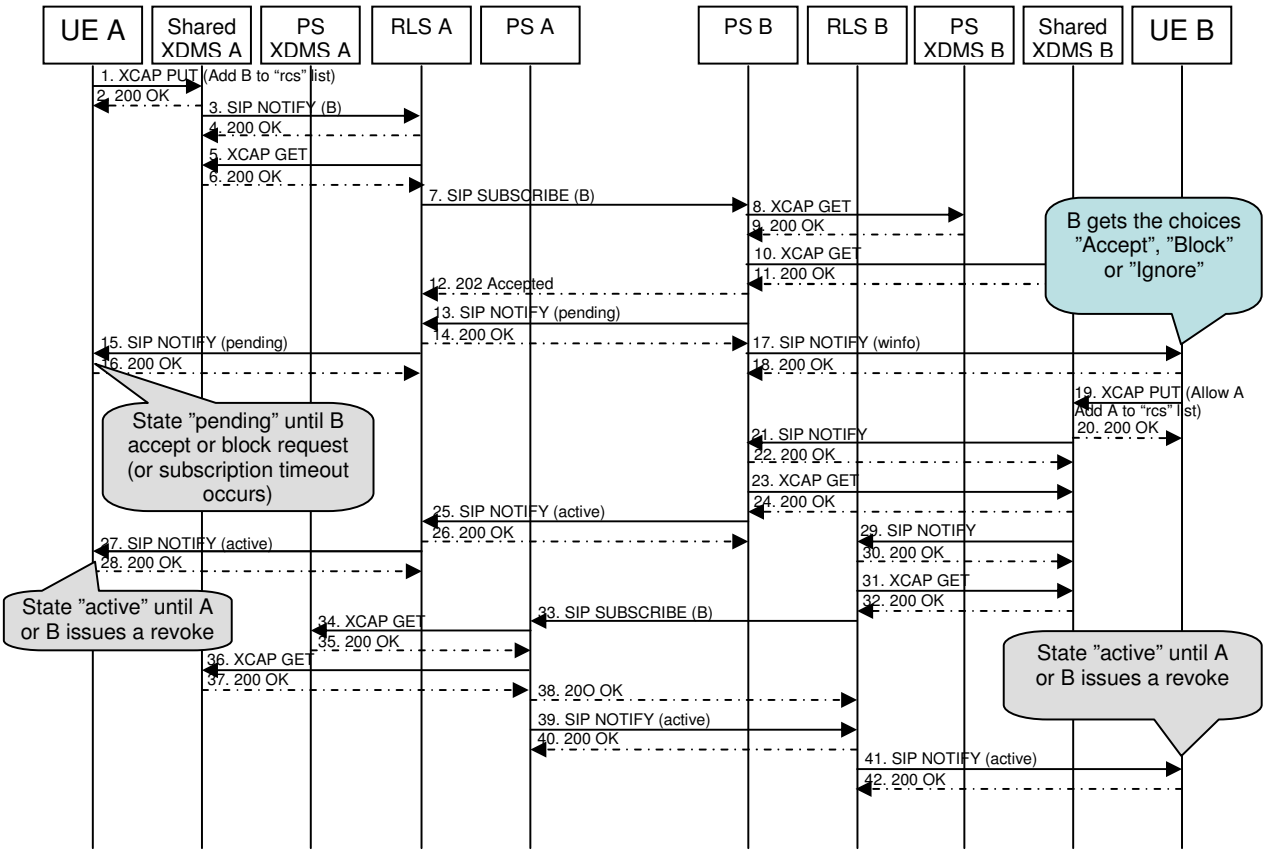
This document concentrates on RCS Release 1 technical realization, so certain features are left for further release(s). For the avoidance of doubt, this does *not* however imply in any way that these features are unimportant, but simply for the interest of time there is a need to limit the first release of RCS .

The following non-exhaustive list gives a high level view of potential items for the later releases of RCS development:

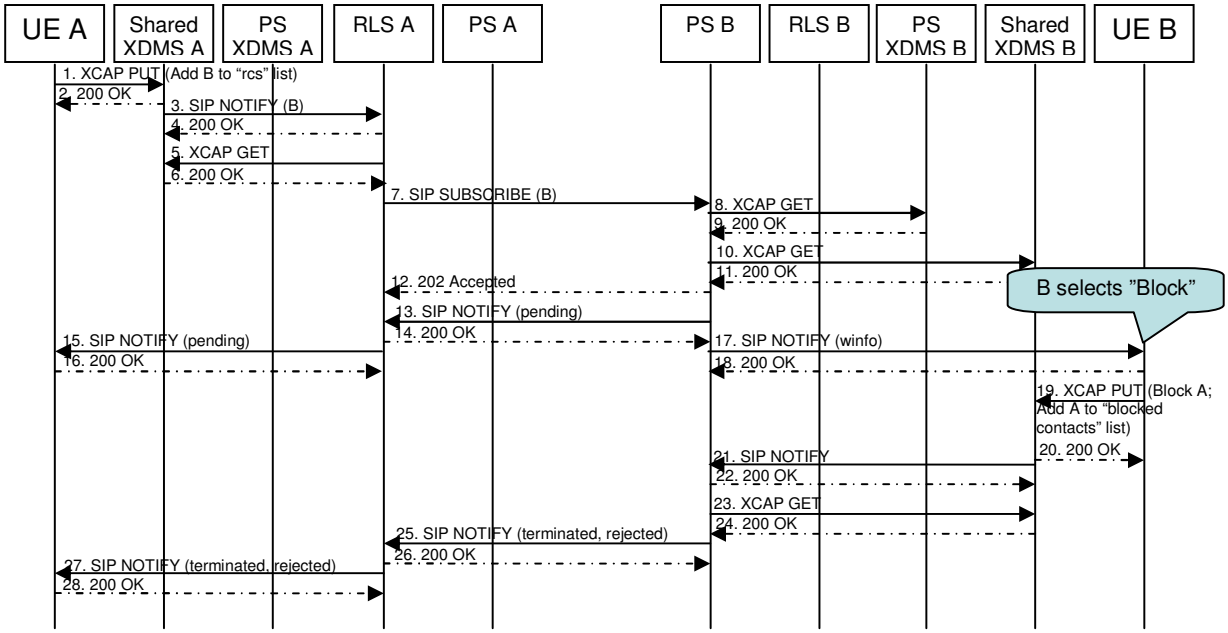
- Usage of RCS services from PC/fixed/broadband access
- Non-(U)SIM based authentication
- Details of network address book
- Presence / capability update optimization issues
- Presence Service Availability and Willingness parameters
- Full compliancy with OMA SIP/SIMPLE IM
- Network side messaging interworking
- Multimedia CLI
- Network based communication log

APPENDIX B: SIGNALLING SEQUENCES FOR PRESENCE SHARING (INFORMATIVE)

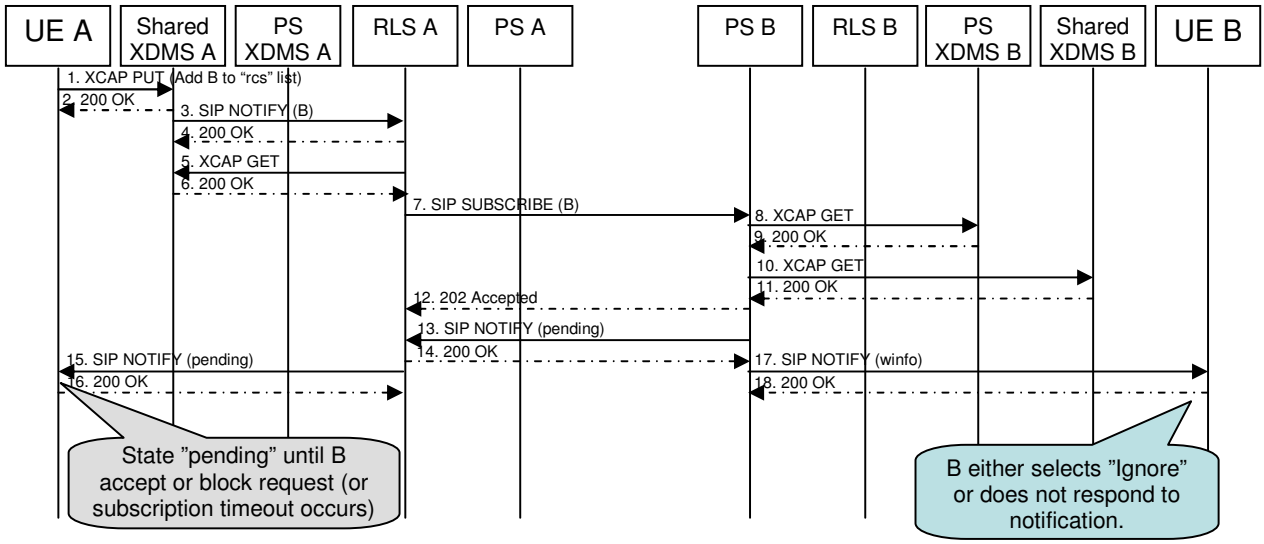
B.1 Symmetric Presence invite and reactive authorization, Accept



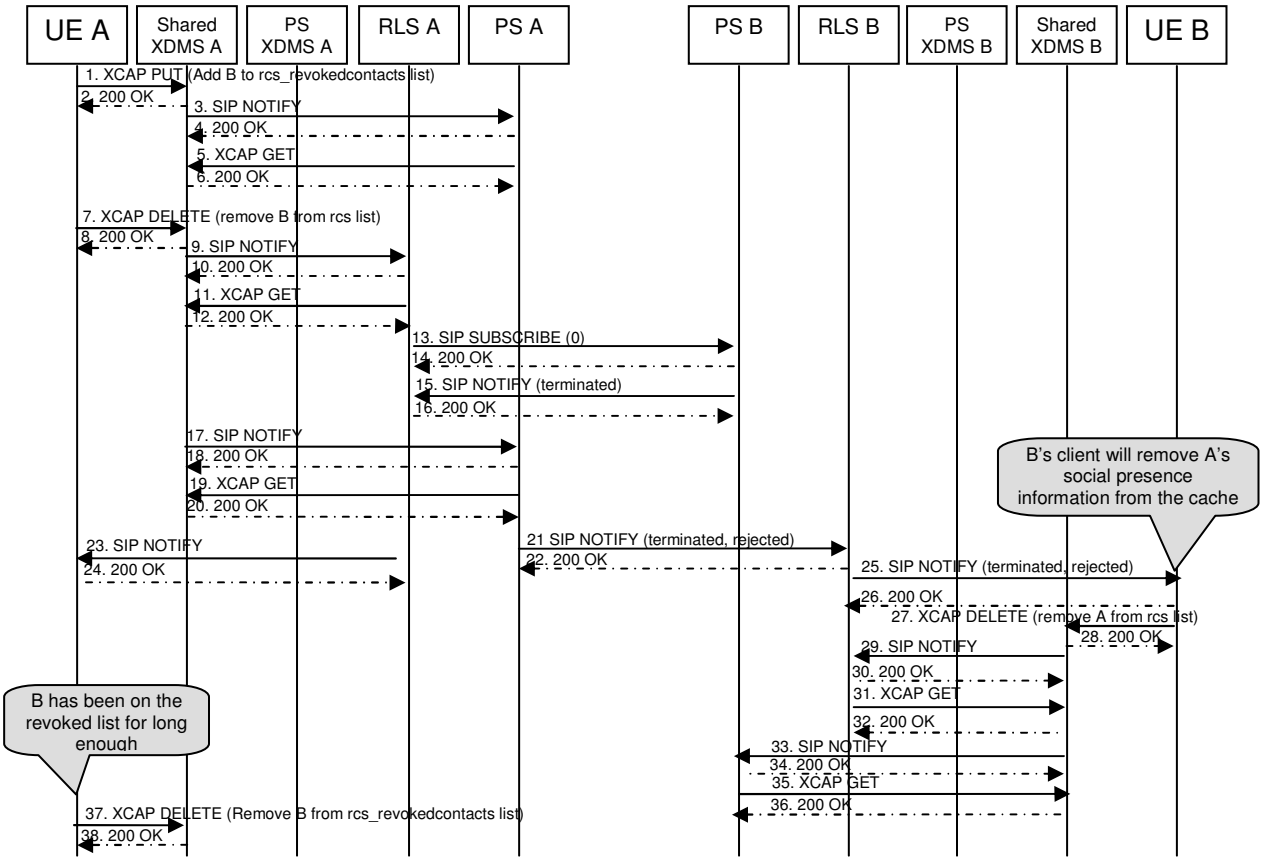
B.2 Symmetric Presence invite and reactive authorization, Block



B.3 Symmetric Presence invite and reactive authorization, Ignore/No Answer



B.4 Terminating a presence relationship using Revoke



DOCUMENT MANAGEMENT

Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
0.1	4 Dec 08	First version using the official GSMA template. Document "RCS Tech Realization v1_021208" used as a basis, including the following CRs: NNI CR (Orange) RLS CR (ALU) Register clarification CR (Ericsson) 2008-TR0002R02 2008-TR0014R2 2008-TR0015R2 2008-TR0016R1 2008-TR0017 2008-TR0018 2008-TR0019 2008-TR0020 2008-TR0025R2 2008-TR0026R4 2008-TR0027R3 2008-TR0028R1 2008-TR0032R1 2008-TR0034 2008-TR0035R1 2008-TR0036 2008-TR0037R2 2008-TR0039R3 2008-TR0041R3 2008-TR0043	RCS Programme	Tero Jalkanen / TeliaSonera
0.2	15 Dec 08	Update based on comments received in Consistency Review	RCS Programme	Tero Jalkanen / TeliaSonera
0.3	16 Dec 08	Grammatical updates in preparation for GSMA QA for DAG approval	Mark Hogan	Mark Hogan/GSMA
1.0	20 Jan 08	Approved by DAG and EMC and updated version to v1.0	DAG & EMC	Mark Hogan/GSMA
1.01	3 June 09	Updated baseline for June edition, including the following CRs: RCS1-BF0003- Conveying the Asserted Identity of an XCAP request - V0 5 RCS1-BF0006R01 - Correction to Appendix B4 RCS1-BF0008 R2_Editorial corrections to RCS R1 TR	RCS Programme	Tero Jalkanen / TeliaSonera

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
		CR_RCS1_BF0011R1_Clarifying the use of emoticon RCS1-BF0014_r1_Clarification_on_service_capabilities_TR RCS-BF0019-Updating RCS 1 Network Address Book Section RCS1 CR CS Voice Capability-RCS1-BF0020 RCS1-BF0021 R1_reject of session invite		
1.02	11 June 09	Update after Berg meeting, including CRs: RCS1 CR Multi-device-presence-RCS1-BF0015R1 RCS1-BF0024 R1 Clarification Addressing Section RCS1-BF0028R2-Aliging NAB section of Func and TR	RCS Programme	
1.03	22 June 09	No Comments received during consistency review R1 : Review report : SPEC DOC RCS SPEC R1_006 https://infocentre.gsm.org/cgi-bin/docindex.cgi?33476 Page 2 added, needed for DAG approval	RCS Programme	Dirk Raeymaekers/ NSN
1.04	25 June 09	Accept changes front pages & grammar/spelling check	RCS Programme	Dirk Raeymaekers /NSN
1.05	12 Aug 09	Remove incorrect reference to R2 in tracking page	RCS Programme	Mark Hogan / GSMA
1.1	31 Aug 09	Approved by DAG & EMC and updated version to V1.1	DAG & EMC	Dirk Raeymaekers /NSN
1.11	16 Sept 09	Updated typo error, approved in TG 06/08/09, no CR needed	RCS Programme	Dirk Raeymaekers /NSN
1.12	25 Nov 09	Update incorporating CRs: RCS1_BF0102R3 Clarification regarding support for wideband speech communication capabilities based on AMR-WB in RCS Release 1 RCS1_BF0103 Clarification regarding support for video codec H264/AVC in RCS Release 1 for video share	RCS Programme	Tero Jalkanen / TeliaSonera
1.13	1 Dec 09	Minor update based on TG San	RCS Programme	Tero Jalkanen

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
		Francisco comments		/ TeliaSonera
1.14	11 DEC09	Update 1.13 (Approved at Plenary 3/12/09) with front pages for DAG approval. No review comments received during consistency review See SPEC DOC RCS SPEC R1_016 in https://infocentre.gsm.org/cgi-bin/docindex.cgi?33476	RCS Programme	Dirk Raeymaekers /NSN
1.2	25th Feb 2010	Approved by DAG/EMC, removal DAG review sheet	RCS Programme	Dirk Raeymaekers /NSN
1.21	18 March 2010	Incorporation of CRs • RCS1-TR-BF0200 R01 Clarifying Hyper Availability • RCS1-TR-BF0201 – R01 Clarification about use of Etag for status icon	RCS Programme	Tero Jalkanen / TeliaSonera
1.22	9 June 2010	Incorporation of CR RCS1-TR-BF204 conditional event notification support	RCS Programme	Tero Jalkanen / TeliaSonera
1.23	15 June 2010	Incorporation of CR RCS1_TR_BF0205 Watcher behaviour clarification when receiving multiple person elements	RCS Programme	Tero Jalkanen / TeliaSonera
1.24	24 JUN10	Update 1.23 (Approved at Plenary 17/6/10) with front pages for DAG approval.	RCS Programme	Dirk Raeymaekers /NSN
1.3	Aug 2010	Approved by DAG/EMC, removal DAG review sheet	RCS Programme	Dirk Raeymaekers /NSN
1.31	1 December 2010	Update after RCS Paris meeting. Editorial correction (chapter numbering) & incorporated CR • RCS1-BF300 Several bug fixes • 2010-TR0211R2 Remove hyper-availability	RCS Programme	Tero Jalkanen / TeliaSonera
1.32	10 December 2010	Editorial consistency update	RCS Programme	Tero Jalkanen / TeliaSonera
2.0	14 February	Submitted to DAG & EMC for approval	EMC	Tero Jalkanen / TeliaSonera

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
	2011			

Other Information

Type	Description
Document Owner	Rich Communication Suite Programme
Editor / Company	Tero Jalkanen / TeliaSonera