



RCS Profile of RESTful Network APIs

Approved Version 1.0 – 16 Apr 2013

Open Mobile Alliance
OMA-TS-REST_NetAPI_RCSPProfile-V1_0-20130416-A

Use of this document is subject to all of the terms and conditions of the Use Agreement located at <http://www.openmobilealliance.org/UseAgreement.html>.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at <http://www.openmobilealliance.org/ipr.html>. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

Contents

1.	SCOPE.....	4
2.	REFERENCES	5
2.1	NORMATIVE REFERENCES.....	5
2.2	INFORMATIVE REFERENCES.....	5
3.	TERMINOLOGY AND CONVENTIONS.....	6
3.1	CONVENTIONS.....	6
3.2	DEFINITIONS.....	6
3.3	ABBREVIATIONS	6
4.	INTRODUCTION	7
4.1	VERSION 1.0	7
5.	RCS PROFILE OF RESTFUL NETWORK APIS.....	8
5.1	NOTIFICATION CHANNEL	8
5.2	NETWORK ADDRESS BOOK.....	8
5.3	PRESENCE.....	9
5.4	MESSAGING	13
5.5	CHAT	13
5.6	FILE TRANSFER.....	16
5.7	THIRD PARTY CALL.....	17
5.8	CALL NOTIFICATION	17
5.9	VIDEO SHARE	18
5.10	IMAGE SHARE.....	19
APPENDIX A.	CHANGE HISTORY (INFORMATIVE).....	21
A.1	APPROVED VERSION HISTORY	21
APPENDIX B.	STATIC CONFORMANCE REQUIREMENTS (NORMATIVE).....	22
B.1	SCR FOR RCS.NOTIFICATIONCHANNEL SERVER.....	22
B.2	SCR FOR RCS.NETWORKADDRESSBOOK SERVER.....	22
B.3	SCR FOR RCS.PRESENCE SERVER.....	23
B.4	SCR FOR RCS.MESSAGING SERVER.....	26
B.5	SCR FOR RCS.CHAT SERVER	27
B.6	SCR FOR RCS.FILETRANSFER SERVER	29
B.7	SCR FOR RCS.THIRDPARTYCALL SERVER	30
B.8	SCR FOR RCS.CALLNOTIFICATION SERVER.....	30
B.9	SCR FOR RCS.VIDEOSHARE SERVER.....	31
B.10	SCR FOR RCS.IMAGESHARE SERVER	32
APPENDIX C.	USING OMA AUTHORIZATION FRAMEWORK FOR NETWORK APIS	34

1. Scope

This specification provides the RCS profile of RESTful Network APIs.

The RCS profile of RESTful Network APIs specifies a subset of the existing OMA RESTful Network APIs. This specification contains tables with information on what operations are mandated in the profile that **MUST** be implemented in order to claim conformance with the profile.

The GSMA RCS project is addressing deployment and operational considerations for 3rd party applications, and is re-using a subset of the OMA RESTful Network APIs for this. It aims to reduce the effort and time needed to create applications and content that is portable across mobile operators.

2. References

2.1 Normative References

- [**Autho4API_10**] “Authorization Framework for Network APIs”, Open Mobile Alliance™, OMA-ER-Autho4API-V1_0, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [**RC_API_RD**] “APIs for Rich Communications Requirements”, Open Mobile Alliance™, OMA-RD-REST_NetAPI_RCSPProfile-V1_0, [URL: http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [**REST_NetAPI_3PC**] “RESTful Network API for Third Party Call”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_ThirdPartyCall-V1_0, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [**REST_NetAPI_AddressBook**] “RESTful Network API for Address Book”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_AddressBook-V1_0, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [**REST_NetAPI_CallNotif**] “RESTful Network API for Call Notification”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_CallNotification-V1_0, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [**REST_NetAPI_Chat**] “RESTful Network API for Chat”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_Chat-V1_0, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [**REST_NetAPI_FileTransfer**] “RESTful Network API for File Transfer”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_FileTransfer-V1_0, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [**REST_NetAPI_ImageShare**] “RESTful Network API for Image Share”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_ImageShare-V1_0, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [**REST_NetAPI_Messaging**] “RESTful Network API for Messaging”, Version 1.0, Open Mobile Alliance™, OMA-TS-REST_NetAPI_Messaging-V1_0, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [**REST_NetAPI_NotifChan**] “RESTful Network API for Notification Channel”, Version 1.0, Open Mobile Alliance™, OMA-TS-REST_NetAPI_NotificationChannel-V1_0, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [**REST_NetAPI_Presence**] “RESTful Network API for Presence”, Version 1.0, Open Mobile Alliance™, OMA-TS-REST_NetAPI_Presence-V1_0, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [**REST_NetAPI_VideoShare**] “RESTful Network API for Video Share”, Version 1.0, Open Mobile Alliance™, OMA-TS-REST_NetAPI_VideoShare-V1_0, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [**RFC2119**] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, [URL:http://www.ietf.org/rfc/rfc2119.txt](http://www.ietf.org/rfc/rfc2119.txt)
- [**SCRRULES**] “SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)

2.2 Informative References

- [**OMADICT**] “Dictionary for OMA Specifications”, Version 2.9, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_9, [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

For the purpose of this TS, all definitions from OMA Dictionary [OMADICT] as well as definitions from individual RESTful Network APIs apply.

3.3 Abbreviations

NAB	Network Address Book
GSMA	GSM Association
HTTP	Hypertext Transfer Protocol
MMS	Multimedia Messaging Service
OMA	Open Mobile Alliance
RCS	Rich Communication Suite
REST	REpresentational State Transfer
SMS	Short Message Service

4. Introduction

The RCS profile of RESTful Network APIs defines a subset of the resources and HTTP methods in these APIs that must be supported by any entity conforming to the profile. The profile does not change the operations themselves in any way, e.g. parameters, whether optional or mandatory, behaviour, etc.

4.1 Version 1.0

Version 1.0 of the RCS Profile of RESTful Network APIs defines subsets of the following APIs:

- RESTful Network API for Notification Channel V 1.0
- RESTful Network API for Address Book V 1.0
- RESTful Network API for Presence V 1.0
- RESTful Network API for Messaging V 1.0
- RESTful Network API for Chat V 1.0
- RESTful Network API for File Transfer V 1.0
- RESTful Network API for Third Party Call V 1.0
- RESTful Network API for Call Notification V 1.0
- RESTful Network API for Video Share V 1.0
- RESTful Network API for Image Share V 1.0

as specified in the following chapter.

5. RCS Profile of RESTful Network APIs

This section gives an overview in a form of tables of all RCS operations and their relations to the RESTful Network APIs.

The section numbers in the column “REST Methods” refer to sections in the RESTful Network APIs specifications where more details about REST methods, operations and examples that relate to RCS operations can be found. The column “Comments” gives some clarifications to the related RCS operations.

5.1 Notification Channel

The RCS profile of the RESTful Network API for Notification Channel defines a subset of the HTTP resources/methods in [REST_NetAPI_NotifChnl] as listed below.

RCS Operations	REST Methods	Comments
Establish Notification Channel for Long Polling	6.1.5	To create Notification Channel, client application uses POST method described in 6.1.5 on the resource defined in 6.1. POST response includes ‘channel URL’ which is used to retrieve notifications using ‘Retrieve notifications from the Notification Server using Long Polling’ operation described below, and ‘callback URL’ that client application should use as notification URL when subscribing to notifications for a particular service.
Retrieve Notification Channel information	6.2.3	To retrieve information about Notification Channel created with ‘Establish Notification Channel for Long Polling’ operation, client application uses GET method described in 6.2.3 for the resource defined in 6.2.
Terminate Notification Channel	6.2.6	To terminate Notification Channel, client application uses DELETE method described in 6.2.3 for the resource defined in 6.2.
Retrieve notifications from Notification Server using Long Polling	6.3.5	To retrieve notifications from the Notification Server, client application uses POST method on the ‘channel URL’ received during the creation of the channel with ‘Establish Notification Channel for Long Polling’ operation.

5.2 Network Address Book

The RCS profile of the RESTful Network API for Address Book defines a subset of the HTTP resources/methods in [REST_NetAPI_AddressBook] as listed below.

RCS Operations	REST Methods	Comments
Retrieve list of contacts in the NAB	6.1.3	Retrieve list of contacts, optionally with all attributes or selected attributes. Note: NAB is “created” when the first contact is added.
Retrieve all information for a specified contact in the VCard	6.2.3	Retrieve information for a contact, optionally with selected

format		attributes. Note: VCard 2.1 and VCard3.0 are retrieved as opaque objects.
Add a new contact to NAB	6.2.4	Add a new contact with or without selected attributes (e.g. VCard).
Update a contact's information	6.2.4, 6.2.5	Update a contact's attributes (6.2.4) or delete a contact (6.2.5).
Subscribe to updates to contacts in NAB	6.15.5	Create a subscription to updates in NAB. MAY include adding and/or removing a contact and/or changing an existing contact's attributes. Note that for client applications that cannot maintain notification URL, the notification URL SHALL be previously obtained by executing the operation 'Establish Notification Channel for Long Polling' (section 5.1).
Receive notifications regarding updates to contacts in NAB	6.17.5	Receive notifications with updates at the notification URL provided when executing 'Subscribe to updates to contacts in NAB' operation. Note that for client applications that cannot maintain a notification URL, the updates SHALL be obtained by executing the 'Retrieve notifications from Notification Server using Long Polling' operation (section 5.1).
Remove subscription	6.16.6	Remove subscription as a temporary resource.

5.3 Presence

The RCS Profile of the RESTful Network API for Presence defines a subset of the HTTP resources/methods in [REST_NetAPI_Presence] as listed below. Note that for some RCS operations related to Presence, HTTP resources/methods are defined in [REST_NetAPI_AddressBook] which is indicated in "REST Methods" column.

RCS Operations	REST Methods	Comments
Manage "free text" presence attribute	6.4.3, 6.4.4, 6.4.6	This operation is used to manage own "free text" social presence attribute. Client application can use GET (retrieve, 6.4.3), PUT (set or update, 6.4.4) and DELETE (remove, 6.4.6) methods defined for the resource in section 6.4. Note that the resource in 6.4 provides access to complete presence attributes data structures, and the "free text" attribute is represented by element 'noteList' from data type 'PersonAttributes' in section 5.2.2.4.
Manage "portrait icon" presence attribute	6.29.3, 6.29.4, 6.29.6	This operation is used to manage own "portrait icon" social presence attribute. The resource that client application shall use is defined in section 6.29. To retrieve "portrait icon", client application uses GET method in section 6.29.3. To set or update "portrait icon", client application uses PUT method in section 6.29.4. This operation will simultaneously upload "portrait

		<p>icon” and set the link to the “portrait icon” as presence information.</p> <p>To remove “portrait icon” social presence attribute, client application uses DELETE method in section 6.29.6.</p>
Manage “favourite link” presence attribute	6.4.3, 6.4.4, 6.4.6	<p>This operation is used to manage own “favourite link” social presence attribute. Client application can use GET (retrieve 6.4.3), PUT (set or update, 6.4.4) and DELETE (remove, 6.4.6) methods defined for the resource in section 6.4.</p> <p>Note that the resource in 6.4 provides access to complete presence attributes data structures, and the “favourite link” attribute is represented by element ‘linkList’ from data type ‘PersonAttributes’ in section 5.2.2.4.</p>
Manage “location” presence attribute	6.4.3, 6.4.4, 6.4.6	<p>This operation is used to manage own “location” social presence attribute. Client application can use GET (retrieve, 6.4.3), PUT (set or update, 6.4.4) and DELETE (remove, 6.4.6) methods defined for the resource in section 6.4.</p> <p>Note that the resource in 6.4 provides access to complete presence attributes data structures, and depending on the format of “location” data, the following elements from data type ‘PersonAttributes’ in section 5.2.2.4 can be used to represent the “location” attribute:</p> <ul style="list-style-type: none"> - ‘placeType’, when location expressed in a form of plain text, - ‘location’ when location expressed with coordinate values, - ‘timeOffset’ when location expressed as time offset from UTC.
Manage “availability status” presence attribute	6.4.3, 6.4.4, 6.4.6	<p>This operation is used to manage own “availability status” social presence attribute. Client application can use GET (retrieve, 6.4.3), PUT (set or update, 6.4.4) and DELETE (remove, 6.4.6) methods defined for the resource in section 6.4.</p> <p>Note that the resource in 6.4 provides access to complete presence attributes data structures, and the “availability status” attribute is represented by element ‘overridingWillingness’ from data type ‘PersonAttributes’ in section 5.2.2.4.</p>
Invite a member to share presence information	6.10.4 from [REST_NetAPI_AddressBook]	<p>Invitation of member to share presence information is done by adding a user (member) to a member list (e.g. ‘rcs’ list).</p> <p>To add a user (member) on the list, client applications use PUT method as described in [REST_NetAPI_AddressBook], section 6.10.4.</p> <p>However this action itself will not trigger presence sharing invitation towards the invited user unless:</p> <ul style="list-style-type: none"> - the inviting user has subscribed for notifications on presence information for that particular member list, as described in ‘Subscribe for notifications on presence information updates for a contact list’ operation, or - the inviting user attempts to retrieve presence information for that particular member list, as described in ‘Retrieve presence information for a contact list’

		Note that in order to see the invitation, the invited user must either subscribe for notifications by executing 'Subscribe for notifications on presence sharing invitation' operation or to retrieve pending invitations by executing 'Retrieve pending invitations' operation.
Cancel presence invitation	6.10.6 from [REST_NetAPI_AddressBook]	To cancel presence sharing invitation before the invitation is accepted, client applications use DELETE method as described in [REST_NetAPI_AddressBook], section 6.10.6
Retrieve presence information for a single contact	6.13.3	To retrieve social presence information for a given contact, client application uses GET method described in section 6.13.3 on the resource defined in 6.13.
Retrieve presence information for a contact list	6.15.3	To retrieve social presence information for all contacts (members) in a contact list, client application uses GET method described in section 6.15.3 on the resource defined in 6.15
Subscribe for notifications on presence sharing invitation	6.18.5	To subscribe for notifications for presence sharing invitation, client application uses POST method described in section 6.18.5 on the resource defined in 6.18. Note that for client applications that cannot maintain notification URL, the notification URL SHALL be previously obtained by executing the operation 'Establish Notification Channel for Long Polling' operation (section 5.1).
Retrieve notifications for presence sharing invitation	6.20.5	Receive notifications for presence sharing invitations at the notification URL provided when executing 'Subscribe for presence sharing invitation' operation. Note that for client applications that cannot maintain a notification URL, the updates SHALL be obtained by executing the 'Retrieve notifications from Notification Server using Long Poling' operation (section 5.1).
Manage presence sharing invitation	6.10.4, 6.27.5 from [REST_NetAPI_AddressBook]	To accept presence sharing invitation from a user, client application uses PUT method described in [REST_NetAPI_AddressBook], section 6.10.4, to add the user to his/her list (e.g. 'rcs' list). To block presence sharing invitation from a user, client application uses PUT method described in [REST_NetAPI_AddressBook], section 6.10.4, to add the user to his/her 'blockedcontacts' list. To revoke presence sharing invitation from a user, client application uses POST method described in [REST_NetAPI_AddressBook], section 6.27.5, and transfer the user from one list to another list (e.g. from 'rcs' list to 'rcs_revokedcontacts' list). Note these operations are used to manage presence sharing relationships only, not as response to invitations for presence sharing. To ignore presence sharing invitation, no specific action is performed by client application. Note that in order to see the invitation, client application (invited user) must either subscribe for notifications by executing 'Subscribe for notifications on presence sharing invitation' operation, or to

		retrieve pending invitations by executing 'Retrieve pending invitations' operation.
Retrieve presence information for own Presentity	6.13.3	To retrieve presence information for the own Presentity, client application uses GET method described in 6.13.3 for the resource defined in 6.13.
Subscribe for notifications on presence information updates for own Presentity	6.22.5	To subscribe for notifications to updates on presence information for own Presentity, client application uses POST method described in section 6.22.5 on the resource defined in 6.22. Note that for client applications that cannot maintain notification URL, the notification URL SHALL be previously obtained by executing the operation 'Establish Notification Channel for Long Polling' (section 5.1).
Subscribe for notifications on presence information updates for a contact list	6.26.5	To subscribe for notifications to updates in presence information for a contact list, client application uses POST method described in section 6.26.5 on the resource defined in 6.26. Note that for client applications that cannot maintain notification URL, the notification URL SHALL be previously obtained by executing the operation 'Establish Notification Channel for Long Polling' (section 5.1).
Retrieve notifications on presence information updates for own Presentity	6.24.5	Receive notifications for presence sharing invitations at the notification URL provided when executing 'Subscribe for updates on presence information for own Presentity' operation. Note that for client applications that cannot maintain a notification URL, the updates SHALL be obtained by executing the 'Retrieve notifications from Notification Server using Long Poling' operation (section 5.1).
Retrieve notifications on presence information updates for a contact list	6.28.5	Receive notifications for presence sharing invitations at the notification URL provided when executing 'Subscribe for updates on presence information for a contact list' operation. Note that for client applications that cannot maintain a notification URL, the updates SHALL be obtained by executing the 'Retrieve notifications from Notification Server using Long Poling' operation (section 5.1).
Retrieve pending invitations	6.8.3	To retrieve pending invitations, client application uses GET method described in section 6.8.3 on the resource defined in 6.8.
Retrieve service capabilities	6.13.3	To retrieve service capabilities (either for own Presentity or for a contact), client application uses GET method described in section 6.13.3 on the resource defined in 6.13, with query parameter set to "?presenceFilter=service/*/*" (without quotation marks)
Publish own service capabilities	6.1.5	To publish own service capabilities, client application uses POST method described in 6.1.5 for the resource defined in 6.1. Data structure 5.2.2.5 describes service capabilities for a particular service.

5.4 Messaging

The RCS Profile of the RESTful Network API for Messaging defines a subset of the HTTP resources/methods in [REST_NetAPI_Messaging] as listed below.

RCS Operations	REST Methods	Comments
Send message	6.9.5	<p>Create an outbound message request. To send an SMS or MMS, to one or multiple destination address(es).</p> <p>To request delivery confirmation via a notification, the client application must provide a notification URL when executing this operation.</p> <p>Note that for client applications that cannot maintain notification URL, the notification URL SHALL be previously obtained by executing the operation 'Establish Notification Channel for Long Polling' (section 5.1).</p>
Receive delivery confirmation	6.14.5	<p>Receive delivery confirmation at the notification URL provided when executing 'Send message' operation.</p> <p>Note that for client applications that cannot maintain a notification URL, the updates SHALL be obtained by executing the 'Retrieve notifications from Notification Server using Long Polling' operation (section 5.1).</p>
Subscribe to receive messages	6.6.5	<p>Create a subscription to receive messages.</p> <p>Note that for client applications that cannot maintain notification URL, the notification URL SHALL be previously obtained by executing the operation 'Establish Notification Channel for Long Polling' (section 5.1).</p>
Receive message(s)	6.8.5	<p>Receive message(s) at the notification URL provided when executing 'Subscribe to receive messages' operation MUST be executed once before this operation.</p> <p>Note that for client applications that cannot maintain a notification URL, the updates SHALL be obtained by executing the 'Retrieve notifications from Notification Server using Long Polling' operation (section 5.1).</p>

5.5 Chat

The RCS Profile of the RESTful Network API for Chat defines a subset of the HTTP resources/methods in [REST_NetAPI_Chat] as listed below.

RCS Operations	REST Methods	Comments
Subscribe for notifications on updates in a chat session	6.1.5	To create a subscription for notifications on updates in a chat session (for example: chat message, chat events, session invitation, participant status, and message delivery status), the

		<p>client application uses the POST method described in 6.1.5 on the resource defined in 6.1.</p> <p>Note that for client applications that cannot expose a notification URL, the notification URL SHALL be previously obtained by executing the operation 'Establish Notification Channel for Long Polling' (section 5.1)</p>
Cancel subscription for notifications on updates in a chat session	6.2.6	To cancel a subscription for notifications on updates in a chat session, the client application uses the DELETE method described in 6.2.6 on the resource defined in 6.2.
Receive notifications on updates in a chat session	6.15.5, 6.16.5, 6.17.5, 6.18.5, 6.19.5, 6.20.5, 6.21.5	<p>Receive notifications on updates in a chat session at the notification URL provided when executing 'Subscribe for notifications on updates in a chat session' operation.</p> <p>For incoming messages, as described in section 6.15.5.</p> <p>For notifications about message status, as described in section 6.16.5</p> <p>For notifications about 1-1 chat session invitation, as described in section 6.17.5.</p> <p>For notifications about a group chat session invitation, as described in section 6.18.5.</p> <p>For notifications about chat session events, as described in section 6.19.5.</p> <p>For notifications about changes in participant status, as described in section 6.20.5.</p> <p>For notifications about subscription termination, as described in section 6.21.5.</p> <p>Note that for client applications that cannot maintain a notification URL, the updates SHALL be obtained by executing the 'Retrieve notifications from Notification Server using Long Polling' operation (section 5.1).</p>
Create (initiate) 1-1 chat session	6.3.5	To initiate 1-1 chat session, the client application uses the POST method described in 6.3.5 on the resource defined in 6.3. The initial chat message is included in the request.
Accept 1-1 chat session invitation	6.5.4	To accept a 1-1 chat session invitation, the client application uses the PUT method described in 6.5.4 on the resource defined in 6.5.
Create (send) 1-1 chat message	6.7.5	To create (send) a new chat message, the client application uses the POST method described in 6.7.5 on the resource defined in 6.7.
Cancel 1-1 chat session invitation	6.4.6	<p>This operation is applicable for the inviting RCS client only. To have an effect, the operation must be performed before the invitation has been accepted.</p> <p>To cancel a 1-1 chat session invitation, the inviting client application uses the DELETE method described in 6.4.6 on the resource defined in 6.4.</p>

Decline 1-1 chat session invitation	6.4.6	This operation is applicable for invited RCS clientonly. To decline (reject) a 1-1 chat session invitation, the invited client application uses the DELETE method described in 6.4.6 on the resource defined in 6.4.
Leave 1-1 chat session	6.4.6	This operation is the same as the following, "Terminate a 1-1 chat session". The reason is that if one participant leaves a 1-1 chat, the chat will end.
Terminate 1-1 chat session	6.4.6	To leave i.e. terminate a 1-1 chat session, the client application uses the DELETE method described in 6.4.6 on the resource defined in 6.4.
Report message status for 1-1 chat session	6.8.4	To report message status, the client application uses the PUT method described in 6.8.4 on the resource defined in 6.8.
Extend 1-1 chat to group chat session	6.6.5	To extend a 1-1 chat session to a group chat session, the client application uses the POST method described in 6.6.5 on the resource defined in 6.6.
Create (initiate) group chat session	6.9.5	To create (initiate) a group chat, the client application uses the POST method described in 6.9.5 on the resource defined in 6.10.
Accept group chat invitation	6.13.4	To accept an invitation for a group chat delivery, the client application uses the POST method described in 6.13.4 on the resource defined in 6.13.
Decline a group chat session invitation	6.12.6	To decline a group chat session invitation, the invited client application uses the DELETE method described in 6.12.6 on the resource defined in 6.12.
Cancel a group chat session invitation	6.10.6	To cancel a group chat session before the first invited participant has accepted the invitation, the originator's application uses the DELETE method described in 6.10.6 on the resource defined in 6.10.
Create (send) group chat message	6.14.5	To create (send) a new group chat message, the client application uses the POST method described in 6.14.5 on the resource defined in 6.14.
Add one or more participants to a group chat session	6.11.5	To add one or more participants to the established group chat session, the client application uses the POST method described in 6.11.5 on the resource defined in 6.11. Note that it depends on a server policy who can add new participants to a group chat session (any Participant or just the Originator)
Remove participant(s) from a group chat session	6.12.6	To remove one or more participants from the established group chat session, the client application uses the DELETE method described in 6.12.6 on the resource defined in 6.12. Note that it depends on a server policy whether this OPTIONAL function is available, and who can remove participant(s) from a group chat session (any Participant or just the Originator)
Leave group chat session	6.12.6	To leave a group chat session, the client application uses the DELETE method described in 6.12.6 on the resource defined in

		6.12.
Re-join a group chat session	6.11.5	To re-join the established group chat session, the client application uses the POST method described in 6.11.5 on the resource defined in 6.11.
Terminate a group chat session	6.10.6	To terminate a group chat session, the Originator's client application uses the DELETE method described in 6.10.6 on the resource defined in 6.10. The availability of this OPTIONAL method depends on service provider policies.

5.6 File Transfer

The RCS Profile of the RESTful Network API for FileTransfer defines a subset of the HTTP resources/methods in [REST_NetAPI_FileTransfer] as listed below.

RCS Operations	REST Methods	Comments
Create a new 1-1 file transfer session	6.3.5	The operation can either include the actual file content or just external file repository URL. The operation can support multi-files in one session, for RCS application there should be only one file.
Cancel file transfer invitation	6.4.6	When the file transfer session status is "Invited", the application of originator can use this operation to cancel file transfer invitation
End the file transfer session	6.4.6	When the file transfer session status is "Connected", both the application of originator and the application of receiver can use this operation to terminate the file transfer session.
Create subscription for file transfer notifications	6.1.5	
Cancel subscription for file transfer notifications	6.2.6	To cancel a subscription for file transfer notifications, the client application uses the DELETE method described in 6.2.6 on the resource defined in 6.2.
Notifications about File Transfer event (declined, cancelled, ended) and MSRP transfer session state "success", "abort" and "error")	6.8.5	The client notification about file transfer session events which include SessionCancelled, SessionEnded, Declined, Successful, Failed, Aborted.
Notification about receiver acceptance	6.10.5	The client notification about receiver acceptance. If receiver accepts the file transfer, the notification includes the status which is "Connected" and the accepted file should be only one file supported in RCS application.
Notification about file transfer invitation	6.7.5	In SessionInvitationNotification, there should be only one file supported in RCS application.
Accept a file transfer invitation	6.5.5	In ReceiverSessionStatus, there should be only one file supported

		in RCS application.
Decline a file transfer invitation	6.4.6	When the file transfer session status is "Invited", the application of receiver can use this operation to decline file transfer invitation.
Notification about file content link	6.9.5	In FileNotification, there should be only one file supported in RCS application.
Notification about subscription cancellation	6.11.5	The client notification that occurs as a result of the cancellation of the subscription by the client.

5.7 Third Party Call

The RCS Profile of the RESTful Network API for Third Party Call defines a subset of the HTTP resources/methods in [REST_NetAPI_3PC] as listed below.

RCS Operations	REST Methods	Comments
Initiate call session	6.1.5	To create call session, client application uses POST method described in 6.1.5 on the resource defined in 6.1
Retrieve call session information	6.2.3	To retrieve call session information, client application uses GET method described in 6.2.3 on the resource defined in 6.2
Terminate call session	6.2.6	To terminate call session, client application uses DELETE method described in 6.2.6 on the resource defined in 6.2

5.8 Call Notification

The RCS Profile of the RESTful Network API for Call Notification defines a subset of the HTTP resources/methods in [REST_NetAPI_CallNotif] as listed below.

RCS Operations	REST Methods	Comments
Subscribe to notifications about call events	6.2.5	To subscribe to notifications about call events, client application uses POST method described in 6.2.5 on the resource defined in 6.2 Section 5.2.3.1 gives details about the call events the client can subscribe for.
Retrieve an individual subscription to notifications about call events	6.3.3	To retrieve an individual subscription to notifications about call events, client application uses GET method described in 6.3.3 on the resource defined in 6.3
Cancel an individual subscription to notifications about call events	6.3.6	To cancel (delete) an individual subscription to notifications about call events, client application uses DELETE method described in 6.3.6 on the resource defined in 6.3
Retrieve notifications about call events	6.10.5	Receive notifications about call events at the notification URL provided when executing 'Subscribe to notifications about call

		<p>events' operation.</p> <p>The updates SHALL be obtained by executing the 'Retrieve notifications from Notification Server using Long Poling' operation (section 5.1).</p>
--	--	--

5.9 Video Share

The RCS Profile of the RESTful Network API for Video Share defines a subset of the HTTP resources/methods in [REST_NetAPI_VideoShare] as listed below.

RCS Operations	REST Methods	Comments
Create subscription to video share notifications	6.1.5	<p>Create a new subscription to video share notifications.</p> <p>Note that for client applications that cannot maintain notification URL, the notification URL SHALL be previously obtained by executing the operation 'Establish Notification Channel for Long Polling' (section 5.1).</p>
Cancel subscription to video share notifications	6.2.6	Cancel a subscription and stop corresponding notifications.
Create a new 1-1 video share session	6.3.5	<p>Create a new 1-1 video share session with CS call related or without CS call related.</p> <p>The operation supports recorded video and live video.</p>
Cancel a 1-1 video share session invitation	6.4.6	Cancel a 1-1 video share session invitation before the invitation has been accepted.
Decline a 1-1 video share session invitation	6.4.6	Decline a 1-1 video share session invitation before the invitation has been accepted.
End a 1-1 video share session	6.4.6	Terminate a 1-1 video share session after the session has been in connected status.
Accept a 1-1 video share session invitation	6.5.5	Accept a 1-1 video share session invitation with supported media format and receive the media URL for accessing the video content.
Receive notifications for video share invitations	6.6.5	<p>Receive notifications for video share session invitations at the notification URL provided when executing 'Create subscription to video share notifications' operation.</p> <p>Note that for client applications that cannot maintain a notification URL, the updates SHALL be obtained by executing the 'Retrieve notifications from Notification Server using Long Polling' operation (section 5.1).</p>
Receive notifications for video share session acceptance	6.7.5	<p>Receive notifications for video share session invitation acceptance at the notification URL provided when executing 'Create subscription to video share notifications' operation.</p> <p>Note that for client applications that cannot maintain a notification URL, the updates SHALL be obtained by executing the 'Retrieve notifications from Notification Server using Long Polling' operation</p>

		(section 5.1).
Receive notifications for video share events	6.8.5	<p>Receive notifications for video share session events at the notification URL provided when executing 'Create subscription to video share notifications' operation.</p> <p>Note that for client applications that cannot maintain a notification URL, the updates SHALL be obtained by executing the 'Retrieve notifications from Notification Server using Long Polling' operation (section 5.1).</p>

5.10 Image Share

The RCS Profile of the RESTful Network API for Image Share defines a subset of the HTTP resources/methods in [REST_NetAPI_ImageShare] as listed below.

RCS Operations	REST Methods	Comments
Create subscription to image share notifications	6.1.5	<p>Create a new subscription to image share notifications.</p> <p>Note that for client applications that cannot maintain notification URL, the notification URL SHALL be previously obtained by executing the operation 'Establish Notification Channel for Long Polling' (section 5.1).</p>
Cancel subscription to image share notifications	6.2.6	Cancel a subscription and stop corresponding notifications.
Create a new image share session	6.3.5	<p>Create a new image share session with CS call related or without CS call related.</p> <p>The operation can either include the actual image file content or include just a file repository URL via which the image file content can be retrieved,</p>
Cancel an image share invitation	6.4.6	Cancel an image share session invitation before the invitation has been accepted.
Decline an image share invitation	6.4.6	Decline an image share session invitation before the invitation has been accepted.
End an image share session	6.4.6	Terminate an image share session after the session has been in connected status.
Accept an image share invitation	6.5.5	Accept an image share session invitation.
Receive notifications for image share invitations	6.6.5	<p>Receive notifications for image share session invitations at the notification URL provided when executing 'Create subscription to image share notifications' operation.</p> <p>Note that for client applications that cannot maintain a notification URL, the updates SHALL be obtained by executing the 'Retrieve notifications from Notification Server using Long Polling' operation (section 5.1).</p>

Receive notifications for image share session acceptance	6.7.5	<p>Receive notifications for image share session invitation acceptance at the notification URL provided when executing 'Create subscription to image share notifications' operation.</p> <p>Note that for client applications that cannot maintain a notification URL, the updates SHALL be obtained by executing the 'Retrieve notifications from Notification Server using Long Polling' operation (section 5.1).</p>
Receive notifications for links to the image file	6.8.5	<p>Receive notifications for links to the image file at the notification URL provided when executing 'Create subscription to image share notifications' operation.</p> <p>Client applications can use the link to download the image file.</p> <p>Note that for client applications that cannot maintain a notification URL, the updates SHALL be obtained by executing the 'Retrieve notifications from Notification Server using Long Polling' operation (section 5.1).</p>
Receive notifications for image share events and image file delivery status.	6.9.5	<p>Receive notifications for image share session events and image file delivery status at the notification URL provided when executing 'Create subscription to image share notifications' operation.</p> <p>Note that for client applications that cannot maintain a notification URL, the updates SHALL be obtained by executing the 'Retrieve notifications from Notification Server using Long Polling' operation (section 5.1).</p>

Appendix A. Change History

(Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-REST_NetAPI_RCSPProfile-V1_0-20130416-A	16 Apr 2013	Status changed to Approved by TP TP Ref # OMA-TP-2013-0103-INP_RC_APIs_V1_0_ERP_for_Final_Approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

B.1 SCR for RCS.NotificationChannel Server

Support for the RCS Profile of the RESTful Network API for Notification Channel implies supporting all mandatory SCRs as defined in the SCR tables of [REST_NetAPI_NotifChnl].

Item	Function	Reference	Requirement
RCS-NOTIFCHNL-S-001-O	Support for the RESTful Notification Channel API	[REST_NetAPI_NotifChnl] Appendix B	REST-NC-SUPPORT-S-001-M AND REST-NC-SUPPORT-S-002-M AND REST-NC-SUPPORT-S-003-M AND REST-NC-CHANNELS-S-001-M AND REST-NC-CHANNELS-S-003-M AND REST-NC-INDCHANNEL-S-001-M AND REST-NC-INDCHANNEL-S-002-M AND REST-NC-INDCHANNEL-S-003-M AND REST-NC-LONGPOLL-S-001-M AND REST-NC-LONGPOLL-S-002-M

B.2 SCR for RCS.NetworkAddressBook Server

Support for the RCS Profile of the RESTful Network API for Address Book implies supporting all mandatory SCRs as defined in the SCR tables of [REST_NetAPI_AddressBook].

Item	Function	Reference	Requirement
RCS-NETAB-S-001-O	Support for the RESTful AddressBook API	[REST_NetAPI_AddressBook] Appendix B	REST-AB-SUPPORT-S-001-M AND REST-AB-SUPPORT-S-002-M AND REST-AB-SUPPORT-S-003-M AND REST-AB-CONTACT-COL-S-001-M AND REST-AB-CONTACT-COL-S-002-M AND REST-AB-CONTACT-IND-S-001-M AND REST-AB-CONTACT-IND-S-002-M AND REST-AB-CONTACT-IND-S-003-M

Item	Function	Reference	Requirement
			AND REST-AB-CONTACT-IND-S-004-M AND REST-AB-ATTRIB-CONTACT-S-001-M AND REST-AB-ATTRIB-CONTACT-S-002-M AND REST-AB-IND-ATTRIB-CONTACT-S-001-M AND REST-AB-IND-ATTRIB-CONTACT-S-002-M AND REST-AB-IND-ATTRIB-CONTACT-S-003-M AND REST-AB-IND-ATTRIB-CONTACT-S-004-M AND REST-AB-SUBSCR-S-001-M AND REST-AB-SUBSCR-S-002-M AND REST-AB-SUBSCR-S-003-M AND REST-AB-IND-SUBSCR-S-001-M AND REST-AB-IND-SUBSCR-S-002-M AND REST-AB-IND-SUBSCR-S-003-M AND REST-AB-IND-SUBSCR-S-004-M AND REST-AB-NOTIF-S-001-M AND REST-AB-NOTIF-S-002-M AND REST-AB-IND-MEM-TRANS-S-001-M AND REST-AB-IND-MEM-TRANS-S-002-M

B.3 SCR for RCS.Presence Server

Support for the RCS Profile of the RESTful Network API for Presence implies supporting all mandatory SCRs as defined in the SCR tables of [REST_NetAPI_Presence].

Item	Function	Reference	Requirement
RCS-PRESENCE-S-001-O	Support for the RESTful Presence API	[REST_NetAPI_Presence] Appendix B	REST-PRESENCE-SUPPORT-S-001-M AND REST-PRESENCE-SUPPORT-S-002-M AND REST-PRESENCE-SUPPORT-S-003-M AND

Item	Function	Reference	Requirement
			REST-PRESENCE-PRES-PS-S-001-M AND REST-PRESENCE-PRES-PS-S-002-M AND REST-PRESENCE-PRES-PS-S-003-M AND REST-PRESENCE-PRES-IND-PS-S-001-M AND REST-PRESENCE-PRES-IND-PS-S-002-M AND REST-PRESENCE-PRES-IND-PS-S-003-M AND REST-PRESENCE-PRES-IND-PS-S-004-M AND REST-PRESENCE-PRES-PRSOURCE-PERS-S-001-M AND REST-PRESENCE-PRES-PS-PERS-S-002-M AND REST-PRESENCE-PRES-PS-PERS-S-003-M AND REST-PRESENCE-PRES-PS-PERS-S-004-M AND REST-PRESENCE-PRES-CONTL-S-001-M AND REST-PRESENCE-PRES-CONTL-S-002-M AND REST-PRESENCE-PRES-IND-CONT-S-001-M AND REST-PRESENCE-PRES-IND-CONT-S-002-M AND REST-PRESENCE-PRES-IND-CONT-S-003-M AND REST-PRESENCE-PRES-IND-CONT-S-004-M AND REST-PRESENCE-PRES-WL-S-001-M AND REST-PRESENCE-PRES-WL-S-002-M AND REST-PRESENCE-PRES-IND-WATCHER-S-001-M AND REST-PRESENCE-PRES-IND-WATCHER-S-002-M AND REST-PRESENCE-PRES-AUTH-RULES-S-001-M AND REST-PRESENCE-PRES-AUTH-RULES-S-002-M

Item	Function	Reference	Requirement
			AND REST-PRESENCE-PRES-AUTH-RULES-S-003-M AND REST-PRESENCE-PRES-IND-AUTH-RULE-S-001-M AND REST-PRESENCE-PRES-IND-AUTH-RULE-S-002-M AND REST-PRESENCE-PRES-IND-AUTH-RULE-S-003-M AND REST-PRESENCE-PRES-IND-AUTH-RULE-S-004-M AND REST-PRESENCE-WATCH-PC-S-001-M AND REST-PRESENCE-WATCH-PC-S-002-M AND REST-PRESENCE-WATCH-PL-S-001-M AND REST-PRESENCE-WATCH-PL-S-002-M AND REST-PRESENCE-WATCH-PCC-S-001-M AND REST-PRESENCE-WATCH-PCC-S-002-M AND REST-PRESENCE-PRES-SUBSCR-WS-S-001-M AND REST-PRESENCE-PRES-SUBSCR-WS-S-003-M AND REST-PRESENCE-PRES-IND-SUBSCR-WS-S-001-M AND REST-PRESENCE-PRES-IND-SUBSCR-WS-S-003-M AND REST-PRESENCE-PRES-IND-SUBSCR-WS-S-004-M AND REST-PRESENCE-WS-NOTIF-S-001-M AND REST-PRESENCE-PRES-WS-S-002-M AND REST-PRESENCE-WATCH-SUBSCR-PS-SINGP-S-001-M AND REST-PRESENCE-WATCH-SUBSCR-PS-SINGP-S-003-M

Item	Function	Reference	Requirement
			AND REST-PRESENCE-WATCH-IND-SUBSCR-PS-SINGP-S-001-M AND REST-PRESENCE-WATCH-IND-SUBSCR-PS-SINGP-S-003-M AND REST-PRESENCE-WATCH-IND-SUBSCR-PS-SINGP-S-004-M AND REST-PRESENCE-PS-NOTIF-S-001-M AND REST-PRESENCE-PS-NOTIF-S-002-M AND REST-PRESENCE-WATCH-SUBSCR-PLS-SINGPL-S-001-M AND REST-PRESENCE-WATCH-SUBSCR-PLS-SINGPL-S-003-M AND REST-PRESENCE-WATCH-IND-SUBSCR-PLS-SINGPL-S-001-M AND REST-PRESENCE-WATCH-IND-SUBSCR-PLS-SINGPL-S-003-M AND REST-PRESENCE-WATCH-IND-SUBSCR-PLS-SINGPL-S-004-M AND REST-PRESENCE-PLS-NOTIF-S-001-M AND REST-PRESENCE-PLS-NOTIF-S-002-M AND REST-PRESENCE-PRES-PORTR-ICON-S-001-M AND REST-PRESENCE-PRES-PORTR-ICON-S-002-M AND REST-PRESENCE-PRES-PORTR-ICON-S-003-M AND REST-PRESENCE-PRES-PORTR-ICON-S-004-M

B.4 SCR for RCS.Messaging Server

Support for the RCS Profile of the RESTful Network API for Messaging implies supporting all mandatory SCRs as defined in the SCR tables of [REST_NetAPI_Messaging].

Item	Function	Reference	Requirement
RCS-MESSAGING-S-001-O	Support for the RESTful Messaging API	[REST_NetAPI_Messaging] Appendix B	REST-MSG-SUPPORT-S-001-M AND REST-MSG-SUPPORT-S-002-M

Item	Function	Reference	Requirement
			AND REST-MSG-SUPPORT-S-003-M AND REST-MSG-OUTB-S-001-M AND REST-MSG-OUTB-S-003-M AND REST-MSG-INB-ONL-SUBSCR-S-001-M AND REST-MSG-INB-ONL-SUBSCR-S-003-M AND REST-MSG-INB-INDON-SUBSCR-S-003-M AND REST-MSG-INB-NOTIF-S-001-M AND REST-MSG-INB-NOTIF-S-002-M AND REST-MSG-OUTB-DELSTAT-NOTIF-S-001-M AND REST-MSG-OUTB-DELSTAT-NOTIF-S-002-M

B.5 SCR for RCS.Chat Server

Support for the RCS Profile of the RESTful Network API for Chat implies supporting those SCRs as defined in the SCR tables of [REST_NetAPI_Chat] that are required by [RC_API_RD].

Item	Function	Reference	Requirement
RCS-CHAT-S-001-O	Support for the RESTful Chat API	[REST_NetAPI_Chat] Appendix B	REST-CHAT-SUPPORT-S-001-M AND REST-CHAT-SUPPORT-S-002-M AND REST-CHAT-SUPPORT-S-003-M AND REST-CHAT-SUBSCR-S-001-M AND REST-CHAT-SUBSCR-S-003-M AND REST-CHAT-SUBSCR-IND-S-001-M AND REST-CHAT-SUBSCR-IND-S-003-M AND REST-CHAT-ONE2ONE-MSG -S-001-M AND REST-CHAT-ONE2ONE-MSG -S-002-M AND REST-CHAT-ONE2ONE-INDMSG-STAT-S-001-M AND REST-CHAT-ONE2ONE-INDMSG-STAT-S-002-M AND

Item	Function	Reference	Requirement
			REST-CHAT-NOTIF MSG-S-001-M AND REST-CHAT-NOTIF-MSG-S-002-M AND REST-CHAT-NOTIF-MSG-STAT-S-001-M AND REST-CHAT-NOTIF-MSG-STAT-S-002-M AND REST-CHAT-GROUP-SESS-S-001-M AND REST-CHAT-GROUP-SESS-S-002-M AND REST-CHAT-GROUP-INDSESS-S-001-M AND REST-CHAT-GROUP-INDSESS-S-003-O AND REST-CHAT-GROUP-INDSESS-S-004-M AND REST-CHAT-GROUP-INDSESS-PART-S-001-M AND REST-CHAT-GROUP-INDSESS-PART-S-003-M AND REST-CHAT-GROUP-INDSESS-INDPART-S-001-M AND REST-CHAT-GROUP-INDSESS-INDPART-S-003-M AND REST-CHAT-GROUP-INDSESS-INDPART-S-001-M AND REST-CHAT-GROUP-INDSESS-INDPART-S-003-M AND REST-CHAT-GROUP-MSG-S-001-M AND REST-CHAT-GROUP-MSG-S-002-M AND REST-CHAT-NOTIF. ONE2ONE-MSG-STAT-S-001-M AND REST-CHAT-NOTIF. ONE2ONE-MSG-STAT-S-002-M AND REST-CHAT-NOTIF-GROUP-INVITE-S-001-M AND REST-CHAT-NOTIF-GROUP-INVITE-S-002-M AND

Item	Function	Reference	Requirement
			REST-CHAT-NOTIF-GROUP-INVITE-S-001-M AND REST-CHAT-NOTIF-GROUP-INVITE-S-002-M AND REST-CHAT-NOTIF-EVENT-S-001-M AND REST-CHAT-NOTIF-EVENT-S-002-M AND REST-CHAT-NOTIF-GROUP-PART-S-001-M AND REST-CHAT-NOTIF-GROUP-PART-S-002-M

B.6 SCR for RCS.FileTransfer Server

Support for the RCS Profile of the RESTful Network API for File Transfer implies supporting all mandatory SCRs as defined in the SCR tables of [REST_NetAPI_FileTransfer].

Item	Function	Reference	Requirement
RCS- FILETRANSFER- S-001-O	Support for the RESTful File Transfer API	[REST_NetAPI_F ileTransfer] Appendix B	REST-FileTransfer-SUPPORT-S-001-M AND REST-FileTransfer-SUPPORT-S-002-M AND REST-FileTransfer-SUPPORT-S-003-M AND REST-FileTransfer-SUBSCR-001-M AND REST-FileTransfer-SUBSCR-003-M AND REST-FileTransfer-IND-SUBSCR-001-M AND REST-FileTransfer-IND-SUBSCR-003-M AND REST-FileTransfer-Sess-001-M AND REST-FileTransfer-Sess-002-M AND REST-FileTransfer-IND-Sess-001-M AND REST-FileTransfer-IND-Sess-003-M AND REST-FileTransfer-IND-Sess-Stat-001-M AND REST-FileTransfer-IND-Sess-Stat-002-M AND REST-FileTransfer-INV-NOTIF-001-M AND REST-FileTransfer-INV-NOTIF-002-M AND REST-FileTransfer-Event-NOTIF-001-M

Item	Function	Reference	Requirement
			AND REST-FileTransfer-Event-NOTIF-002-M AND REST-FileTransfer-Link-NOTIF-001-M AND REST-FileTransfer-Link-NOTIF-002-M AND REST-FileTransfer-RA-NOTIF-001-M AND REST-FileTransfer-RA-NOTIF-002-M Note: SCR for REST.FileTransfer.Session.Files Server is out of RC-API requirement scope.

B.7 SCR for RCS.ThirdPartyCall Server

Support for the RCS Profile of the RESTful Network API for ThirdParty Call implies supporting those SCRs defined in the SCR tables of [REST_NetAPI_3PC] that are required by [RC_API_RD].

Item	Function	Reference	Requirement
RCS-3PC-S-001-O	Support for the RESTful ThirdParty Call API	[REST_NetAPI_3PC] Appendix B	REST-3PC-SUPPORT-S-001-M AND REST-3PC-SUPPORT-S-002-M AND REST-3PC-SUPPORT-S-003-M AND REST-3PC-SESS-S-001-M AND REST-3PC-SESS-S-003-M AND REST-3PC-SESS-S-005-M AND REST-3PC-INDSESS-S-001-M AND REST-3PC-INDSESS-S-002-M AND REST-3PC-INDSESS-S-003-M

B.8 SCR for RCS.CallNotification Server

Support for the RCS Profile of the RESTful Network API for Call Notification implies supporting those SCRs as defined in the SCR tables of [REST_NetAPI_CallNotif] that are required by [RC_API_RD].

Item	Function	Reference	Requirement
RCS-CALLNOTIF-S-001-O	Support for the RESTful Call Notification API	[REST_NetAPI_CallNotif] Appendix B	REST-CN-SUPPORT-S-001-M AND REST-CN-SUPPORT-S-002-M AND REST-CN-SUPPORT-S-003-M AND REST-CN-SUBSCR-CALLEVENT-S-001-M

Item	Function	Reference	Requirement
			AND REST-CN-SUBSCR-CALLEVENT-S-003-M AND REST-CN-SUBSCR-INDCALLEVENT-S-001-M AND REST-CN-SUBSCR-INDCALLEVENT-S-002-M AND REST-CN-SUBSCR-INDCALLEVENT-S-003-M AND REST-CN-NOTIF-CALLEVENT-S-001-M AND REST-CN-NOTIF-CALLEVENT-S-002-M

B.9 SCR for RCS.VideoShare Server

Support for the RCS Profile of the RESTful Network API for Video Share implies supporting all mandatory SCRs as defined in the SCR tables of [REST_NetAPI_VideoShare].

Item	Function	Reference	Requirement
RCS-VIDEOSHARE-S-001-O	Support for the RESTful Video Share API	[REST_NetAPI_VideoShare] Appendix B	REST-VIDEOSHARE-SUPPORT-S-001-M AND REST-VIDEOSHARE-SUPPORT-S-002-M AND REST-VIDEOSHARE-SUPPORT-S-003-M AND REST-VIDEOSHARE-SUBSCR-S-001-M AND REST-VIDEOSHARE-SUBSCR-S-003-M AND REST-VIDEOSHARE-IND-SUBSCR-S-001-M AND REST-VIDEOSHARE-IND-SUBSCR-S-003-M AND REST-VIDEOSHARE-SESS-S-001-M AND REST-VIDEOSHARE-SESS-S-002-M. AND REST-VIDEOSHARE-IND-SESS-S-001-M AND REST-VIDEOSHARE-IND-SESS-S-003-M AND REST-VIDEOSHARE-IND-SESS-STAT-S-001-M AND REST-VIDEOSHARE-IND-SESS-STAT-S-002-M AND REST-VIDEOSHARE-INVITE-NOTIF-S-001-M AND REST-VIDEOSHARE-INVITE-NOTIF-S-002-M AND

Item	Function	Reference	Requirement
			REST-VIDEOSHARE-ACCEPT-NOTIF-S-001-M AND REST-VIDEOSHARE-ACCEPT-NOTIF-S-002-M AND REST-VIDEOSHARE-EVENT-NOTIF-S-001-M AND REST-VIDEOSHARE-EVENT-NOTIF-S-002-M

B.10 SCR for RCS.ImageShare Server

Support for the RCS Profile of the RESTful Network API for Image Share implies supporting all mandatory SCRs as defined in the SCR tables of [REST_NetAPI_ImageShare].

Item	Function	Reference	Requirement
RCS-IMAGESHARE-S-001-O	Support for the RESTful Image Share API	[REST_NetAPI_ImageShare] Appendix B	REST-IMAGESHARE-SUPPORT-S-001-M AND REST-IMAGESHARE-SUPPORT-S-002-M AND REST-IMAGESHARE-SUPPORT-S-003-M AND REST-IMAGESHARE-SUBSCR-S-001-M AND REST-IMAGESHARE-SUBSCR-S-003-M AND REST-IMAGESHARE-IND-SUBSCR-S-001-M AND REST-IMAGESHARE-IND-SUBSCR-S-003-M AND REST-IMAGESHARE-SESS-S-001-M AND REST-IMAGESHARE-SESS-S-002-M. AND REST-IMAGESHARE-IND-SESS-S-001-M AND REST-IMAGESHARE-IND-SESS-S-003-M AND REST-IMAGESHARE-IND-SESS-STAT-S-001-M AND REST-IMAGESHARE-IND-SESS-STAT-S-002-M AND REST-IMAGESHARE-INVITE-NOTIF-S-001-M AND REST-IMAGESHARE-INVITE-NOTIF-S-002-M AND REST-IMAGESHARE-ACCEPT-NOTIF-S-001-M AND REST-IMAGESHARE-ACCEPT-NOTIF-S-002-M AND REST-IMAGESHARE-LINK-NOTIF-S-001-M

Item	Function	Reference	Requirement
			AND REST-IMAGESHARE-LINK-NOTIF-S-002-M AND REST-IMAGESHARE-EVENT-NOTIF-S-001-M AND REST-IMAGESHARE-EVENT-NOTIF-S-002-M

Appendix C. Using OMA Authorization Framework for Network APIs

All RESTful Network API in this profile MAY support the authorization framework defined in [Autho4API_10]. For details on supported scope values for a specific RESTful Network API see Appendix G of that RESTful Network API (e.g. for scope values supported by [REST_NetAPI_Chat], see [REST_NetAPI_Chat] Appendix G).