



RCS-e Profile of RESTful Network APIs

Candidate Version 1.0 – 05 Mar 2013

Open Mobile Alliance
OMA-TS-REST_NetAPI_RCSeProfile-V1_0-20130305-C

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-E PROFILE OF RESTFUL NETWORK APIS..... | 8 |
| 5.1 | NOTIFICATION CHANNEL | 8 |
| 5.2 | CHAT | 8 |
| 5.3 | FILE TRANSFER..... | 10 |
| 5.4 | THIRD PARTY CALL | 12 |
| 5.5 | CALL NOTIFICATION | 13 |
| 5.6 | VIDEO SHARE | 13 |
| 5.7 | IMAGE SHARE..... | 14 |
| 5.8 | ANONYMOUS CUSTOMER REFERENCE MANAGEMENT | 15 |
| 5.9 | CAPABILITY DISCOVERY | 16 |
| APPENDIX A. | CHANGE HISTORY (INFORMATIVE)..... | 18 |
| A.1 | APPROVED VERSION HISTORY | 18 |
| A.2 | DRAFT VERSION 1.0 HISTORY | 18 |
| APPENDIX B. | STATIC CONFORMANCE REQUIREMENTS (NORMATIVE)..... | 19 |
| B.1 | SCR FOR RCSE.NOTIFICATIONCHANNEL SERVER..... | 19 |
| B.2 | SCR FOR RCSE.CHAT SERVER | 19 |
| B.3 | SCR FOR RCSE.FILETRANSFER SERVER | 21 |
| B.4 | SCR FOR RCSE.THIRDPARTYCALL SERVER..... | 22 |
| B.5 | SCR FOR RCSE.CALLNOTIFICATION SERVER..... | 22 |
| B.6 | SCR FOR RCSE.VIDEOSHARE SERVER | 23 |
| B.7 | SCR FOR RCSE.IMAGESHARE SERVER..... | 24 |
| B.8 | SCR FOR RCSE.ACR SERVER | 25 |
| B.9 | SCR FOR RCSE.CAPABILITYDISCOVERY SERVER..... | 25 |
| APPENDIX C. | USING OMA AUTHORIZATION FRAMEWORK FOR NETWORK APIS | 27 |

1. Scope

This specification provides the RCS-e profile of RESTful Network APIs.

The RCS-e 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 RCE 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/>
- [**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/>
- [**REST_NetAPI_ACR**] RESTful Network API for Anonymous Customer Reference Management”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_ACR-V1_0, URL: <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/>
- [**REST_NetAPI_CapabilityDiscovery**] “RESTful Network API for Capability Discovery”, Version 1.0, Open Mobile Alliance™, OMA-TS-REST_NetAPI_CapabilityDiscovery -V1_0, URL: <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/>
- [**REST_NetAPI_FileTransfer**] “RESTful Network API for File Transfer”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_FileTransfer-V1_0, URL: <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/>
- [**REST_NetAPI_NotifChannel**] “RESTful Network API for Notification Channel”, Version 1.0, Open Mobile Alliance™, OMA-TS-REST_NetAPI_NotificationChannel-V1_0, URL: <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/>
- [**REQ_RCSe**] “Rich Communication Ecosystem RCS-e Network API Detailed Requirements 1.0”, URL: <http://www.gsma.com/rcs/wp-content/uploads/2012/03/rcsenetworkapiv10.pdf>
- [**RFC2119**] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, URL: <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/>

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/>

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

| | |
|-------------|-----------------------------------|
| ACR | Anonymous Customer Reference |
| API | Application Programming Interface |
| GSM | GSM Association |
| HTTP | HyperText Transfer Protocol |
| OMA | Open Mobile Alliance |
| RCE | Rich Communication Ecosystem |
| RCS | Rich Communication Suite |
| REST | REpresentational State Transfer |
| SCR | Static Conformance Requirements |
| TS | Technical Specification |

4. Introduction

The RCS-e 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 as required by GSMA RCE [REQ_RCSe]. 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-e Profile of RESTful Network APIs defines subsets of the following APIs:

- RESTful Network API for Notification Channel 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
- RESTful Network API for Anonymous Customer Reference Management V1.0
- RESTful Network API for Capability Discovery V1.0

as specified in the following chapter.

5. RCS-e Profile of RESTful Network APIs

This section gives an overview in a form of tables of all RCS-e API 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-e operations can be found. The column “Comments” gives some clarifications to the related RCS-e operations.

5.1 Notification Channel

The RCS-e 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-e 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 6.2.6 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 Chat

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

For 1-1 chats, only the ad-hoc variant is supported by RCS-e.

| RCS-e Operations | REST Methods | Comments |
|--|--------------|---|
| Subscribe for notifications on updates in a chat | 6.1.5 | To create a subscription for notifications on updates in a chat (for example: new events, group chat session invitation, participant status, and message delivery status), client application uses POST method described in 6.1.5 on the resource defined in 6.1. |

| | | |
|--|-------------------|--|
| | | <p>For RCS-e, the element “adhocChatSupported” is not allowed to be set to “false”.</p> <p>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 | 6.2.6 | To cancel a subscription for notifications on updates in a chat, client application uses DELETE method described in 6.2.6 on the resource defined in 6.2. |
| Create (send) a 1-1 chat message | 6.7.5 | <p>To send a chat message in an ad-hoc 1-1 chat, client application uses the POST method described in section 6.7.5 to pass a “chatMessage” structure to the resource defined in 6.7.</p> <p>Note that the {sessionId} resource URL variable needs to be set to “adhoc” in the RCS-e profile.</p> |
| Create (send) an “isComposing” message in a 1-1 chat | 6.7.5 | <p>To send an “isComposing” message in an ad-hoc 1-1 chat, client application uses the POST method described in section 6.7.5 to pass an “isComposing” structure to the resource defined in 6.7.</p> <p>Note that the {sessionId} resource URL variable needs to be set to “adhoc” in the RCS-e profile.</p> |
| Confirm the display of a 1-1 chat message | 6.8.4 | <p>To report the displaying of a message in an ad-hoc 1-1 chat, client application uses the PUT method described in section 6.8.4 on the resource defined in 6.8.</p> <p>Note that the {sessionId} resource URL variable needs to be set to “adhoc” in the RCS-e profile.</p> |
| Receive notifications on updates in a 1-1 chat | 6.15.5, 6.16.5 | <p>Receive notifications on updates in an ad-hoc 1-1 chat at the notification URL provided when executing ‘Subscribe for notifications on updates in a chat’ operation.</p> <p>For receiving incoming chat messages, as described in section 6.15.5, with a “chatMessage” structure.</p> <p>For receiving incoming “isComposing” messages, as described in section 6.15.5, with an “isComposing” structure.</p> <p>For receiving notifications about message status (such as “Delivered”, “Displayed”) in an ad-hoc 1-1 chat, as described in section 6.16.5</p> <p>The updates SHALL be obtained by executing the ‘Retrieve notifications from Notification Server using Long Polling’ operation (section 5.1).</p> |
| Create (initiate) group chat session | 6.9.5 | To create (initiate) a group chat delivery, client application uses POST method described in 6.9.5 on the resource defined in 6.9. |
| Accept group chat invitation | 6.13.4 | To accept an invitation for a group chat delivery, client application uses PUT 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 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 invitation, the inviting client application uses DELETE method described in 6.10.6 on the resource defined in 6.10. |

| | | |
|--|---|--|
| | | To have an effect, the operation must be performed before the invitation has been accepted by some of the invited participants. |
| 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, client application uses 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, client application uses DELETE method described in 6.12.6 on the resource defined in 6.12. Note that it depends on a server policy who can remove participant(s) from a group chat session (any Participant or just the Originator). Note further that this is an optional feature. |
| Leave group chat session | 6.12.6 | To remove one or more participants from the established group chat session, client application uses DELETE method described in 6.12.6 on the resource defined in 6.12. Note that it depends on a server policy what will happen with the group chat session if the Originator leaves the session (e.g. to terminate session or continue the session with the rest of participants) |
| Re-join a group chat session | 6.11.5 | To re-join the established group chat session, client application uses POST method described in 6.11.5 on the resource defined in 6.11 |
| Create (send) group chat message | 6.14.5 | To send a chat message in a group chat, client application uses the POST method described in section 6.14.5 to pass a “chatMessage” structure to the resource defined in 6.14. |
| Receive notifications on updates in a group chat | 6.15.5, 6.18.5, 6.19.5, 6.20.5 | Receive notifications on updates in a group chat session at the notification URL provided when executing ‘Subscribe for notifications on updates in a chat’ operation. For receiving incoming chat messages, as described in section 6.15.5, with a “chatMessage” structure. For notifications about a group chat session invitation, as described in section 6.18.5. For notifications about chat session events such as SessionEnded, SessionCancelled, as described in section 6.19.5. For notifications about changes in participant status, as described in section 6.20.5. The updates SHALL be obtained by executing the ‘Retrieve notifications from Notification Server using Long Polling’ operation (section 5.1). |

5.3 File Transfer

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

| RCS-e Operations | REST Methods | Comments |
|---|--------------|--|
| Create a new 1-1 file transfer session | 6.3.5 | <p>To create a 1-1 file transfer session, client application uses POST method described in 6.3.5 on the resource defined in 6.3.5. The operation can either include the actual file content or just external file repository URL.</p> <p>The operation can support multi-files in one session; for RCS-e only one file is supported.</p> |
| Cancel file transfer invitation | 6.4.6 | <p>To cancel a file transfer session invitation, client application uses DELETE method described in 6.4.6 on the resource defined in 6.4.</p> <p>When the file transfer session status is “Invited”, the application of originator can use this operation to cancel file transfer invitation.</p> |
| End the file transfer session | 6.4.6 | <p>To end a file transfer session, client application uses DELETE method described in 6.4.6 on the resource defined in 6.4.</p> <p>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.</p> |
| Create subscription for file transfer notifications | 6.1.5 | <p>To create a subscription for event notifications in a file transfer session, client application uses POST method described in 6.1.5 on the resource defined in 6.1.</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 for file transfer notifications | 6.2.6 | <p>To cancel a subscription for file transfer events notifications, the client application uses the DELETE method described in 6.2.6 on the resource defined in 6.2.</p> |
| Notifications about File Transfer event (declined, cancelled, ended) and MSRP transfer session state (“success”, “abort” and “error”) | 6.8.5 | <p>Receive notifications about file transfer session events at the notification URL provided when executing ‘Create subscription for file transfer notifications’ operation.</p> <p>Events covered by those notifications include SessionCancelled, SessionEnded, Declined, Successful, Failed, Aborted.</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> |
| Notification about receiver acceptance | 6.10.5 | <p>Receive notifications about receiver acceptance for the file transfer session invitation at the notification URL provided when executing ‘Create subscription for file transfer notifications’ operation.</p> <p>If receiver accepts the file transfer, the notification includes the status, which is “Connected”, and for RCSe there should be only one file accepted.</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> |
| Notification about file transfer | 6.7.5 | <p>Receive notifications about file transfer session invitations at the notification URL provided when executing ‘Create subscription for file</p> |

| | | |
|--|--------|---|
| invitation | | <p>transfer notifications' operation.</p> <p>In SessionInvitationNotificaiton, for RCSe there should be only one file included.</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> |
| Accept a file transfer invitation | 6.5.5 | <p>To accept an invitation for file transfer session, client application uses POST method described in 6.5.5 on the resource defined in 6.5.</p> <p>In ReceiverSessionStatus, for RCSe there should be only one file supported.</p> |
| Decline a file transfer invitation | 6.4.6 | <p>To decline an invitation for file transfer session, client application uses DELETE method described in 6.4.6 on the resource defined in 6.4.</p> <p>When the file transfer session status is "Invited", the application of receiver can use this operation to decline file transfer invitation.</p> |
| Notification about file content link | 6.9.5 | <p>Receive notifications about the link to file content at the notification URL provided when executing 'Create subscription for file transfer notifications' operation.</p> <p>In FileNotification, for RCSe there should be only one file supported.</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> |
| Notification about subscription cancellation | 6.11.5 | <p>Receive notifications about subscription cancellations at the notification URL provided when executing 'Create subscription for file transfer 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> |

5.4 Third Party Call

The RCS-e 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-e 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 |

5.5 Call Notification

The RCS-e 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-e 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. 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 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 events' operation. The updates SHALL be obtained by executing the 'Retrieve notifications from Notification Server using Long Poling' operation (section 5.1). |

5.6 Video Share

The RCS-e 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-e Operations | REST Methods | Comments |
|--|--------------|---|
| Create subscription to video share notifications | 6.1.5 | Create a new subscription to video share notifications. 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). |
| 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 | Create a new 1-1 video share session with CS call related or without CS call related. The operation supports recorded video and live video. |
| Cancel a 1-1 video share session | 6.4.6 | Cancel a 1-1 video share session invitation before the invitation has been |

| | | |
|--|-------|--|
| invitation | | 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 | Receive notifications for video share session invitations at the notification URL provided when executing 'Create subscription to video share notifications' 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). |
| Receive notifications for video share session acceptance | 6.7.5 | Receive notifications for video share session invitation acceptance at the notification URL provided when executing 'Create subscription to video share notifications' 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). |
| Receive notifications for video share events | 6.8.5 | Receive notifications for video share session events at the notification URL provided when executing 'Create subscription to video share notifications' 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). |

5.7 Image Share

The RCS-e 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-e Operations | REST Methods | Comments |
|--|--------------|---|
| Create subscription to image share notifications | 6.1.5 | Create a new subscription to image share notifications. 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). |
| 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 | Create a new image share session with CS call related or without CS call related. 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, |
| 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 conncted status. |
| Accept an image share invitation | 6.5.5 | Accept an image share session invitation. |
| Receive notifications for image share invitations | 6.6.5 | Receive notifications for image share session invitations at the notification URL provided when executing 'Create subscription to image share notifications' 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).. |
| Receive notification s for image share session acceptance | 6.7.5 | Receive notifications forimage share session invitation acceptance at the notification URL provided when executing 'Create subscription to image share notifications' 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). |
| Receive notifications for links to the image file | 6.8.5 | Receive notifications for links to the image file at the notification URL provided when executing 'Create subscription to image share notifications' operation. Client applications can use the link to download the image file. 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). |
| Receive notifications for image share events and image file delivery status. | 6.9.5 | 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. 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). |

5.8 Anonymous Customer Reference Management

The RCS-e Profile of the RESTful Network API for Anonymous Customer Reference Management defines a subset of the HTTP resources/methods in [REST_NetAPI_ACR] as listed below.

| RCS-e Operations | REST Methods | Comments |
|--|--------------|---|
| Retrieve the already issued app-specific ACR | 6.1.3 | To retrieve the already issued application-specific ACR for the given end user, client application uses GET method described in |

| | | |
|---|-------|--|
| | | 6.1.3 on the resource defined in 6.1 |
| Requesting ACR associated with a particular mobile telephone number (MSISDN). | 6.1.5 | To issue (create) an application-specific ACR for the given user identified by the userId, client application uses POST method described in 6.1.5 on the resource defined in 6.1 |
| Retrieves the ACR data | 6.2.3 | To retrieve the ACR data (value, status, expiry), client application uses GET method described in 6.2.3 on the resource defined in 6.2 |
| Delete (remove) an ACR | 6.2.6 | To remove an ACR with data, client application uses DELETE method described in 6.2.6 for the resource defined in 6.2. |
| Query the ACR status of an user | 6.3.3 | To retrieve the status of an ACR, client application uses GET method described in 6.3.3 for the resource defined in 6.3. |
| Refresh the user's ACR | 6.3.4 | To refresh an "expired" ACR (i.e. set the ACR status to "valid"), client application uses PUT method described in 6.3.4 on the resource defined in 6.3. |

5.9 Capability Discovery

The RCS-e Profile of the RESTful Network API for Capability Discovery defines a subset of the HTTP resources/methods in [REST_NetAPI_CapabilityDiscovery] as listed below.

| RCS-e Operations | REST Methods | Comments |
|---|---------------------------|--|
| Register a new service capability | 6.1.5, 6.2.4, 6.3.4 | <p>To be able to register a new own service capability, a client application MUST ensure that Capability Source for that particular device/application is created. To create a new Capability Source, the client application uses POST method described in 6.1.5 on the resource defined in 6.1. With the same operation for creating the Capability Source, the application can also register one or more service capabilities for that particular Capability Source.</p> <p>In case the Capability Source has already been created, to register new service capability (ies) the client application SHALL use PUT method described in 6.2.4 on the resource defined in 6.2.</p> <p>Alternatively, for registering a single service capability, client application can use PUT method described in 6.3.4 on the Light-weight Resource defined in 6.3. Note that this is an optional feature.</p> <p>Note that in case a user is registering service capabilities from multiple devices/applications (subject to server policy), in order to manage service capabilities independently for each device/application, the user SHALL create a separate Capability Source for each device/application.</p> |
| Deregister a previously registered service capability | 6.2.4, 6.3.6, 6.2.6 | <p>To deregister (unregister) previously registered own service capability (ies) for a particular Capability Source, client application uses PUT method described in 6.2.4 on the resource defined in 6.2.</p> <p>Alternatively, for deregistering a single service capability, client application can use DELETE method described in 6.3.6 on the Light-weight Resource defined in 6.3. Note that this is an optional</p> |

| | | |
|--|---------------------------|--|
| | | <p>feature.</p> <p>Note: To deregister all service capabilities for a particular Capability Source and at the same time remove the Capability Source completely, client application uses DELETE method described in 6.2.6 on the resource defined in 6.2. When performing this operation the capabilities should be disabled first.</p> |
| Enable or disable a previously registered service capability | 6.2.4, 6.3.4 | <p>To enable or disable previously registered own service capability (ies) for a particular Capability Source, client application uses PUT method described in 6.2.4 on the resource defined in 6.2.</p> <p>Alternatively, for enabling or disabling a single service capability, client application can use PUT method described in 6.3.4 on the Light-weight Resource defined in 6.3. Note that this is an optional feature.</p> |
| Query service capabilities for a contact | 6.4.3 | To query for a service capabilities for a contact, client application uses GET method described in 6.4.3 for the resource defined in 6.4 |
| Query if a certain contact is RCSe capable or not. | 6.4.3 | To query if a certain contact is RCSe capable or not, client application uses GET method described in 6.4.3 for the resource defined in 6.4. In this case, query string parameter 'userTypeFilter' SHALL be used and its value SHALL be set to "RCSe". |
| Retrieve registered own service capabilities | 6.1.3, 6.2.3, 6.3.3 | <p>To retrieve all own service capabilities (registered for all Capability Sources), client application used GET method described in 6.1.3 on the resource defined in 6.1.</p> <p>To retrieve own service capabilities registered for a particular Capability Source, client application uses GET method described in 6.2.3 on the resource defined in 6.2.</p> <p>Alternatively, to retrieve a single own service capability registered for a particular Capability Source, client application uses GET method described in 6.3.3 on the Light-weight Resource defined in 6.3. Note that this is an optional feature.</p> |

Appendix A. Change History (Informative)

A.1 Approved Version History

| Reference | Date | Description |
|-----------|------|------------------|
| n/a | n/a | No prior version |

A.2 Draft Version 1.0 History

| Document Identifier | Date | Sections | Description |
|--|---------------|-----------------------|---|
| Draft Versions OMA-TS-REST_NetAPI_RCSeProfile-V1_0 | 13 March 2012 | all | First draft baseline based on OMA-ARC-2012-0057- INP_BaselineRCSeProfileTS |
| | 11 Nov 2012 | all | Incorporated: OMA-ARC-RCSe-Profile-2012-0006 OMA-ARC-RCSe-Profile-2012-0007 OMA-ARC-RCSe-Profile-2012-0008R01 |
| | 13 Nov 2012 | all | Incorporated: OMA-ARC-RCSe-Profile-2012-0011 OMA-ARC-RCSe-Profile-2012-0009 |
| | 27 Nov 2012 | 2.2, 5.8, 5.9, B.9 | Incorporated: OMA-ARC-RCSe-Profile-2012-0012R02 |
| | 30 Jan 2013 | 2.2, 5.8, 5.9, B.9 | Incorporated: MA-ARC-RCSe-Profile-2013-0002R01- CR_TS_addressing_majority_of_CONRR_comments |
| | 20 Feb 2013 | 2.2, 5.8, 5.9, B.9 | Incorporated: OMA-ARC-RCSe-Profile-2013-0007- CR_TS_addressing_CONRR_comments_for_FT OMA-ARC-RCSe-Profile-2013-0008- CR_TS_updates_covering_ACR |
| Candidate Version OMA-TS-REST_NetAPI_RCSeProfile-V1_0 | 05 Mar 2013 | n/a | Status changed to Candidate by TP TP Ref # OMA-TP-2013-0075- INP_REST_NetAPI_RCSeProfile_V1.0_ERP_and_ETR_for_Candid ate_approval |

Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for RCSe.NotificationChannel Server

Support for the RCS-e 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 |
|------------------------|--|---------------------------------------|--|
| RCSe-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 RCSe.Chat Server

Support for the RCS-e Profile of the RESTful Network API for Chat implies supporting all mandatory SCRs as defined in the SCR tables of [REST_NetAPI_Chat].

| Item | Function | Reference | Requirement |
|-------------------|----------------------------------|----------------------------------|--|
| RCSe-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-SESS-S-001-M |

| Item | Function | Reference | Requirement |
|------|----------|-----------|---|
| | | | AND REST-CHAT-ONE2ONE-INDSESS-ADH-S-001-O AND REST-CHAT-ONE2ONE-INDSESS-EXT-S-001-O AND REST-CHAT-ONE2ONE-INDSESS-EXT-S-002-O 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 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-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 MSG-S-001-M AND REST-CHAT-NOTIF-MSG-S-002-M AND REST-CHAT-NOTIF-MSG-STAT-S-001-M |

| Item | Function | Reference | Requirement |
|-------------------|--------------------------------------|----------------------------------|--|
| | | | AND REST-CHAT-NOTIF-MSG-STAT-S-002-M AND 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 AND |
| RCSe-CHAT-S-002-O | Support for chat session termination | [REST_NetAPI_Chat] Appendix B | REST-CHAT-GROUP-INDSESS-S-003-O |

B.3 SCR for RCSe.FileTransfer Server

Support for the RCS-e 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 |
|---------------------------|---|--|---|
| RCSe-FILETRANSFER-S-001-O | Support for the RESTful File Transfer API | [REST_NetAPI_FileTransfer] 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 |

| Item | Function | Reference | Requirement |
|------|----------|-----------|--|
| | | | 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 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 |

B.4 SCR for RCSe.ThirdPartyCall Server

Support for the RCS-e Profile of the RESTful Network API for ThirdParty Call implies supporting all mandatory SCRs as defined in the SCR tables of [REST_NetAPI_3PC].

| Item | Function | Reference | Requirement |
|------------------|---|---------------------------------|---|
| RCSe-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-INDSESS-S-001-M AND REST-3PC-INDSESS-S-002-M AND REST-3PC-INDSESS-S-003-M |

B.5 SCR for RCSe.CallNotification Server

Support for the RCS-e Profile of the RESTful Network API for Call Notification implies supporting all mandatory SCRs as defined in the SCR tables of [REST_NetAPI_CallNotif].

| Item | Function | Reference | Requirement |
|------------------------|---|---------------------------------------|---|
| RCSe-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 |

| Item | Function | Reference | Requirement |
|------|----------|-----------|---|
| | | | AND REST-CN-SUPPORT-S-003-M AND REST-CN-SUBSCR-CALLEVENT-S-001-M 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.6 SCR for RCSe.VideoShare Server

Support for the RCS-e 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 |
|-------------------------|---|-------------------------------------|---|
| RCSe-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 |

| Item | Function | Reference | Requirement |
|------|----------|-----------|--|
| | | | AND REST-VIDEOSHARE-INVITE-NOTIF-S-001-M AND REST-VIDEOSHARE-INVITE-NOTIF-S-002-M AND 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.7 SCR for RCSe.ImageShare Server

Support for the RCS-e 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 |
|-------------------------|---|-------------------------------------|---|
| RCSe-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 |

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

B.8 SCR for RCSe.ACR Server

Support for the RCS-e Profile of the RESTful Network API for Anonymous Customer Reference Management implies supporting all mandatory SCRs as defined in the SCR tables of [REST_NetAPI_ACR].

| Item | Function | Reference | Requirement |
|-------------------|---|------------------------------|--|
| RCSe-ACRM-S-001-O | Support for the RESTful Anonymous Customer Reference Management API | [REST_NetAPI_ACR] Appendix B | REST-ACRMGNT-SUPPORT-S-001-M AND REST-ACRMGNT-SUPPORT-S-002-M AND REST-ACRMGNT-SUPPORT-S-003-M AND REST-ACRMGNT-USERSAPP-S-001-M AND REST-ACRMGNT-USERSAPP-S-002-M AND REST-ACRMGNT-USERSAPP-S-003-M AND REST-ACRMGNT-ACR-S-001-M AND REST-ACRMGNT-ACR-S-002-M AND REST-ACRMGNT-ACR-S-003-M AND REST-ACRMGNT-ACRSTATUS-S-001-M AND REST-ACRMGNT-ACRSTATUS-S-002-M AND REST-ACRMGNT-ACRSTATUS-S-003-M |

B.9 SCR for RCSe.CapabilityDiscovery Server

Support for the RCS-e Profile of the RESTful Network API for CapabilityDiscovery implies supporting all mandatory SCRs as defined in the SCR tables of [REST_NetAPI_CapabilityDiscovery].

| Item | Function | Reference | Requirement |
|---------------------|---|--|------------------------------------|
| RCSe-CAPDIS-S-001-O | Support for the RESTful CapabilityDiscovery | [REST_NetAPI_CapabilityDiscovery] Appendix B | REST-CAPDIS-SUPPORT-S-001-M AND |

| Item | Function | Reference | Requirement |
|------|----------|-----------|---|
| | API | | REST-CAPDIS-SUPPORT-S-002-M AND REST-CAPDIS-SUPPORT-S-003-M AND REST-CAPDIS-OWN-CAPSRC-S-001-M AND REST-CAPDIS-OWN-CAPSRC-S-002-M AND REST-CAPDIS-OWN-CAPSRC-S-003-M AND REST-CAPDIS-OWN-CAPS-S-001-M AND REST-CAPDIS-OWN-CAPS-S-002-M AND REST-CAPDIS-OWN-CAPS-S-003-M AND REST-CAPDIS-OWN-CAPS-S-004-M AND REST-CAPDIS-OWN-IND-CAP-S-001-O AND REST-CAPDIS-OWN-IND-CAP-S-002-O AND REST-CAPDIS-OWN-IND-CAP-S-003-O AND REST-CAPDIS-OWN-IND-CAP-S-004-O AND REST-CAPDIS-CONTACT-CAPS-S-001-M AND REST-CAPDIS-CONTACT-CAPS-S-002-M AND REST-CAPDIS-CONTACT-CAPS-S-004-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).