RCS Profile of ParlayREST Web Services

Approved Version 1.0 – 24 Jul 2012

Open Mobile Alliance

OMA-TS-ParlayREST_RCSProfile-V1_0-20120724-A

Contents

1. Scope

This specification provides the RCS (Rich Communications Suite) profile of the RESTful bindings of Parlay X Web Services [REST_ERP].

2. References

2.1 Normative References

[REST_TS_3PC] "Restful bindings for Parlay X Web Services – Third Party Call", Open Mobile Alliance™, OMA-TS-ParlayREST-ThirdPartyCall-V1_0, URL: http://www.openmobilealliance.org/

[REST_TS_AddressList Mgmt] "Restful bindings for Parlay X Web Services – Address List Management", Open Mobile Alliance™, OMA-TS-ParlayREST-AddressListManagement-V1_0, URL:http://www.openmobilealliance.org/

[REST_TS_AudioCall] "Restful bindings for Parlay X Web Services – AudioCall", Open Mobile Alliance™, OMA-TS-ParlayREST-AudioCall-V1_0, URL: http://www.openmobilealliance.org/

[REST_TS_CallNotif] "Restful bindings for Parlay X Web Services – Call Notification", Open Mobile Alliance™, OMA-TS-ParlayREST-CallNotification-V1_0, URL: http://www.openmobilealliance.org/

[REST_TS_Messaging] "Restful bindings for Parlay X Web Services – Multi-mediaMessaging", Open Mobile Alliance™, OMA-TS-ParlayREST-MultiMediaMessaging-V1_1, URL: http://www.openmobilealliance.org/

[REST_TS_Presence] "Restful bindings for Parlay X Web Services – Presence", Open Mobile Alliance™, OMA-TS-ParlayREST-Presence-V1_0, URL: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

[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 2.8, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_8, URL: http://www.openmobilealliance.org/

[REST_ERP] "RESTful bindings for Parlay X Web Services", Version 2.0, Open Mobile Alliance™, OMA-ERP-ParlayREST-V2_0, 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

(none in this revision)

3.3 Abbreviations

API	Application Programming Interface
DTMF	Dual Tone Multiple Frequency
GSMA	GSM Association
HTTP	Hypertext Transfer Protocol
IS	Image Share
NW	Network
OMA	Open Mobile Alliance
RCS	Rich Communications Suite
REST	REpresentational State Transfer
RLS	Resource List Server
SPI	Social Presence Information
UNI	User to Network Interface
VS	Video Share

4. Introduction

The GSMA RCS project aims to speed up and facilitate the adoption of applications and services that provide an interoperable, convergent, rich communication experience based on IP and Multimedia. In response to the evolving needs of consumers, operators, vendors and developers RCS will use the Application Programming Interface (API) approach in order to expand capabilities, enable new business models, provide mechanisms for differentiated services, and significantly reduce time-to-market for innovation.

The RCS Profile of ParlayREST specifies a subset recommended to be used by GSMA RCS.

This RCS Profile specification contains tables with information on what interfaces are mandated in the profile that must be implemented in order to claim that the instance complies with the profile. The RCS Profile of ParlayREST defines a subset of the resources and HTTP operations in ParlayREST 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

The first version of the RCS profile contains tables for Third Party Call, Call Notification, Audio Call, Presence, Messaging and Address List Management as specified in the following chapter.

5. RCS Profile of ParlayREST (Informative)

5.1 Call Control

The RCS profile of ParlayREST Call Control defines a subset of the HTTP operations in [REST_TS_3PC], [REST_TS_CallNotif] and [REST_TS_AudioCall] as listed below.

Note: The ParlayREST Third Party Call API only includes the control path, not the media path. That means, the API can be used to initiate / terminate a call session between a number of call participants (the exact limitation depends on operator policies; however, it is safe to assume that this number is at least 2). However, the API has no access to the media that make up the call. These are exchanged directly between the terminals (or telephony software clients) of the call participants.

RCS Requirement	ParlayREST Resource	ParlayREST Operation	Comments
Initiate Call	http://{serverRoot}/{apiVersion}/thirdpartycall/callSessions	POST [REST_TS_3PC], 5.4.5	Create call session, optionally subscribing to notifications regarding session progress
	http://{serverRoot}/{apiVersion}/thirdpartycall/callSessions/{callSessionId}/participants	GET [REST_TS_3PC], 5.7.3	Option 1: Poll for session progress by checking the status of the participants (basic)
	(client-defined during session creation)	POST [REST_TS_CallNotif], 5.13.5	Option 2: Receive session progress notifications (advanced)
Cancel Call	http://{serverRoot}/{apiVersion}/thirdpartycall/callSessions/{callSessionId}	DELETE [REST_TS_3PC], 5.5.6	Terminate a call session
Receive Call	http://{serverRoot}/{apiVersion}/callnotification/subscriptions/callEvent	POST [REST_TS_CallNotif], 5.5.5	Subscribe to call event notifications with the event “CalledNumber”.
	(client-defined during subscription)	POST [REST_TS_CallNotif], 5.13.5	Receive notifications about incoming calls
	http://{serverRoot}/{apiVersion}/callnotification/subscriptions/callEvent/{subscriptionId}	DELETE [REST_TS_CallNotif], 5.6.6	Terminate the subscription when no longer interested
Call Decline	http://{serverRoot}/{apiVersion}/callnotification/subscriptions/callDirection	POST [REST_TS_CallNotif], 5.7.5	Subscribe to call direction notifications with the event “CalledNumber”.
	(client-defined during during subscription)	POST [REST_TS_CallNotif], 5.15.5	Receive notifications about incoming calls, and respond with the action “EndCall”, to decline the call on behalf of the user

	http://{serverRoot}/{apiVersion}/callnotification/subscriptions/callDirection/{subscriptionId}	DELETE [REST_TS_CallNotif], 5.8.6	Terminate the subscription when no longer interested
Call Answer	http://{serverRoot}/{apiVersion}/callnotification/subscriptions/callEvent	POST [REST_TS_CallNotif], 5.5.5	Subscribe to call event notifications with the event “Answer”.
	(client-defined during subscription)	POST [REST_TS_CallNotif], 5.13.5	Receive notifications about calls answered.
	http://{serverRoot}/{apiVersion}/callnotification/subscriptions/callEvent/{subscriptionId}	DELETE [REST_TS_CallNotif], 5.6.6	Terminate the subscription when no longer interested
End Call			Same as Cancel Call
Call Alerting	http://{serverRoot}/{apiVersion}/callnotification/subscriptions/callEvent	POST [REST_TS_CallNotif], 5.5.5	Subscribe to call event notifications with the event “CalledNumber”.
	(client-defined during subscription)	POST [REST_TS_CallNotif], 5.13.5	Receive notifications about attempted calls
	http://{serverRoot}/{apiVersion}/callnotification/subscriptions/callEvent/{subscriptionId}	DELETE [REST_TS_CallNotif], 5.6.6	Terminate the subscription when no longer interested
Send Audio	http://{serverRoot}/{apiVersion}/audiocall/messages/audio	POST [REST_TS_AudioCall], 5.8.5	Initiate the playback of an audiomessage to one or more participants in the call
	http://{serverRoot}/{apiVersion}/audiocall/messages/audio/{messageId}/statusList	GET [REST_TS_AudioCall], 5.9.3	Optionally, poll the status of the message for all participants
Receive Audio	http://{serverRoot}/{apiVersion}/callnotification/subscriptions/recording	POST [REST_TS_CallNotif], 5.11.5	Subscribe to notifications to receive the result of the recording
	http://{serverRoot}/{apiVersion}/audiocall/interactions/recording	POST [REST_TS_AudioCall], 5.20.5	Initiate the playback of a prompt message, and the recording of the participants’ audio
	(client-defined during subscription)	POST [REST_TS_CallNotif], 5.14.5	Receive notifications with pointers to recorded media for the participants
	http://{serverRoot}/{apiVersion}/callnotification/subscriptions/recording/{subscriptionId}	DELETE [REST_TS_CallNotif], 5.12.6	Terminate the subscription when after receiving all notifications

5.2 Messaging

The RCS profile of ParlayREST Multi-media Messaging defines a subset of the HTTP operations in [REST_TS_Messaging] as listed below.

Note: The ParlayREST Multi-media Messaging covers both SMS and MMS.

RCS Requirement	ParlayREST Resource	ParlayREST Operation	Comments
Send Message and Receive Delivery Confirmation	http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests	POST [REST_TS_Messaging], 5.12.5	Create create new outbound message request. The requestId identifying the resource is returned as part of the created resource path. Message can be sent to multiple addresses. Outbound message content can be set to SMS or MMS. For delivery confirmation see Option 1 and Option 2 below.
	http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}/deliveryInfos	GET [REST_TS_Messaging], 5.13.3	Option 1: Poll to read delivery status for the individual outbound message request
	(client-defined as part of the send message request)	POST by Server toward application [REST_TS_Messaging], 5.17.5.	Option 2: Receive delivery confirmation via a notification. To use this option, application must provide a callback reference including a notification URL, as part of the Send Message request.
Receive Message(s)	http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages OR http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId} AND	GET [REST_TS_Messaging], 5.4.3 OR [REST_TS_Messaging], 5.7.3	Read one or more inbound messages. For accessing the messages see Option 1 and 2 below.

	(client-defined during the subscription to receive messages)	POST [REST_TS_Messaging], 5.	Option 1: application has previously created a subscription to receive messages. Application will receive notifications that include the resource path for each of the received messages (e.g. http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId}) at the notification URL provided during the subscribtion.
			Option 2: application has not created a subscription to receive messages. The registrationId must be provided off-line in order to retrieve all messages, and the messageId must be provided off-line in order to retrieve a specific message only (outside the scope of the API).
Read Message Attachment	http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId}/attachments/{attachmentId}	GET [REST_TS_Messaging], 5.8.3	Read one MMS message attachment, after having read an MMS message that includes one or more identified attachments (see Read Message(s)).
Subscribe to Receive Messages	http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions	POST [REST_TS_Messaging], 5.13.3	Create a subscription to inbound message, by providing a callback reference including a notification URL. See Receive Message(s). The subscriptionId identifying the resource is returned as part of the created resource path.

Delete a Specific Message	http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId}	DELETE [REST_TS_Messaging], 5.7.6	Delete one particular inbound message (see Read Specific Message).
Delete a Specific Message Attachment	http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId}/attachments/{attachmentId}	DELETE [REST_TS_Messaging], 5.8.6	Delete one particular message (see Read Specific Message Attachment).

5.3 Presence

The RCS profile of ParlayREST Presence defines a subset of the HTTP operations in [REST_TS_Presence] as listed below. The section numbers in the column "ParlayREST Operations" refer to sections in [REST_TS_Presence].

RCS Requirement	ParlayREST Resource	ParlayREST Operation	Comments
Set tagline	http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSources/persistent/person/noteList	PUT 5.8.4	PUT, GET, DELETE are used to manage the persistent data.
Set presence image	http://{serverRoot}/{apiVersion}/presence/{userId}/content/{picture.jpg}	PUT 5.11.4	Upload image/picture
	http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSources/persistent/person/statusIcon	PUT 5.8.4	Update Etag
Set Link	http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSources/persistent/person/linkList	PUT 5.8.4	PUT, GET, DELETE are used to manage the persistent data.
Set location	http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSources/persistent/person/location	PUT 5.8.4	PUT, GET, DELETE are used to manage the persistent data.
Set Willingness	http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSources/persistent/overriding-willingness	PUT 5.8.4	PUT, GET, DELETE are used to manage the persistent data.
Set service capabilities	http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSource	POST 5.4.5	The Server Capabilities are set by sending a POST request in order to create a Presence Source with a specific duration.
Receive presence sharing invitation notification	http://{serverRoot}/{apiVersion}/presence/{userId}/subscriptions/watchersSubscriptions	POST 5.22.5	There are two alternatives to receive updates about a new presence invitation. The first one is that Watcher client subscribes for

			changes in presence sharing invitation Notification are generated using POST (requesting entity provides the call back URL).
	http://{serverRoot}/{apiVersion}/presence/{userId}/watchers	GET 5.12.3	The second alternative to receive updates about presence sharing invitation is to use polling mechanism.
Presence notification of Presence changes	http://{serverRoot}/{apiVersion}/presence/{userId}/subscriptions/presenceListSubscriptions/{memberListId}	POST 5.30.5	There are two alternatives to receive Presence updates. The first one is that Watcher subscribe for notifications for Presence updates by creating subscription for Presence list: Notification are generated using POST (requesting entity provides the call back URL).
	http://{serverRoot}/{apiVersion}/presence/{userId}/presenceLists/{memberListId}	GET 5.19.3	The second alternative for Watcher to receive Presence updates is to use polling mechanism towards the presence contact list.
Request for Service Capabilities (‘who can I invite’)	http://{serverRoot}/{apiVersion}/presence/{userId}/presenceContact/{presentityUserId}	GET 5.17.3	The Watcher client uses polling mechanism for updates by retrieving presence data about the Presentity.

5.4 Address List Management

The RCS profile of ParlayREST Address List Management defines a subset of the HTTP operations in [REST_TS_AddressListMgmt] as listed below. The section numbers in the column “ParlayREST Operations” refer to sections in [REST_TS_ AddressListMgmt].

RCS Requirement	ParlayREST Resource	ParlayREST Operation	Comments
Invite a contact to share presence	http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListI	PUT	By adding an additional user to the rcs-list it will

	d}/members/{memberId}	5.13.4	trigger a presence invitation towards the other party.
Cancel invitation	http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/members/{memberId}	DELETE 5.13.6	By removing the user from the rcs-list it will trigger a cancellation of the presence invitation.
Accept presence sharing invitation	http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/members/{memberId}	PUT 5.13.4	Authorizing a presence invitation is done by adding the user to the rcs-list (mutual authorization).
Block presence sharing invitation	http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/members/{memberId}	PUT 5.13.4	Blocking is done by adding the user to the rcs_blockedcontacts list. See also “Revoke presence sharing invitation” below.
Ignore presence sharing invitation	-	-	No action.
Revoke presence sharing invitation	http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/members/{memberId}	PUT 5.13.4	Revoking of the invitation is done by adding the user to the rcs_revokedcontacts list.
Subscribe to watcher info & RLS lists	-	-	Please refer to ‘Receive presence sharing invitation notification’ in section 5.3.
Management of contact lists	-	-	Please refer to ‘Invite a contact to share presence’ above.
Presence Rules Management	-	-	Please refer to ‘Accept presence sharing invitation’ and ‘Revoke presence sharing invitation’ above.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-ParlayREST_RCSProfile-V1_0-20120724-A	24 Jul 2012	Status changed to Approved by TP Ref TP Doc# OMA-TP-2012-0280-INP_ParlayREST_2_0_for_Final_Approval

Appendix B. Static Conformance Requirements (Normative)

For Static Conformance Requirements please refer to the Static Conformance Requirements for the selected operations, as specified in Appendix B of the ParlayREST Technical Specifications listed as normative references in section 2.1 of this document.