Mobile Augmented Reality Enabler Techinical Specifications

Candidate Version 1.0 – 16 Oct 2012

Open Mobile Alliance

OMA-TS-MobAR-V1_0-20121016-C

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.

Contents

Figures

Tables

1. Scope

This Technical Specification (TS) document defines the components procedures and functionalities as well as operations and parameters of interfaces of the Mobile Augmented Reality Enabler.

This document complements the [MobAR- ER] that provides the requirements and architecture of the Mobile Augmented Reality Enabler.

2. References

2.1 Normative References

[RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

[RFC4234] “Augmented BNF for Syntax Specifications: ABNF”. D. Crocker, Ed., P. Overell. October 2005, URL:http://www.ietf.org/rfc/rfc4234.txt

[SCRRULES] “SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:http://www.openmobilealliance.org/

[MobAR-ER] “Mobile Augmented Reality Enabler Release Document” , Open Mobile Alliance™, OMA-ER- MobAR-V1_0, URL:http://www.openmobilealliance.org/

[RFC2616] “Hypertext Transfer Protocol – HTTP/1.1, R.Fielding et al, June 1999, URL:http://www.ietf.org/rfc/rfc2616.txt

[PUSH-OTA] “Push Over The Air”, Open Mobile Alliance™, OMA-TS-PushOTA-V2_3, URL:http://www.openmobilealliance.org/

[PUSH-PAP] “Push Access Protocol”, Open Mobile Alliance™, OMA-TS-PAP-V2_3, URL:http://www.openmobilealliance.org/

[PUSH-MSG] “Push Message”, Open Mobile Alliance™, OMA-TS-Push_Message-V2_3, URL:http://www.openmobilealliance.org/

[PUSH-SL] “ServiceLoading”, WAP Forum™, WAP-168-ServiceLoad-20010731-a, URL:http://www.openmobilealliance.org/

[RFC2617] “HTTP Authentication: Basic and Digest Access Authentication” URL: http://www.ietf.org/rfc/rfc2617.txt

[OMA TLS] “OMA TLS Profile”, Open Mobile AllianceTM, OMA-TS-TLS_Profile-V1_1 URL:http://www.openmobilealliance.org/

[REST_NetAPI_Common] “Common definitions for RESTful Network APIs”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_Common-V1_0, URL: http://www.openmobilealliance.org/

[XSD-MobAR] “XML Schema for Mobile Augmented Reality”, Open Mobile Alliance™, OMA-SUP-XSD_MobAR-V1_0, URL: http://www.openmobilealliance.org/

[RFC4627] “The application/json Media Type for JavaScript Object Notation (JSON)”

URL: http://www.ietf.org/rfc/rfc4627.txt

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.

Explanation of the XML Type in MobAR tables is as follows:

- E = Elements
- A = Attributes
- 1,2,3, etc = nesting level of the element
- Attributes apply to the parent Element.

Each message start with a message element, noted as E.

Explanation of the Parameter cardinality in MobAR Tables is as follow:

- 1 means the parameter has one instance
- 0…1 means the parameter is either missing or has exactly one occurrence
- 0...N means the parameter has zero or more occurrences.
- 1...N means the parameter has one or more occurrences.

Informative illustration of a table:

Parameter name	XML Type	Data type	Parameter cardinality	Description
Message Name	E			Indicates the name of the message.
Name	E1	String	1	Indicates the name of the element.
SubName	A	Byte	0...N	Indicate the possible subNames of the Name element.

3.2 Definitions

AR Content Any kind of multimedia object which can be used to augment/enhance the user’s sensory perception of the world such as: picture, video, text, 3D models, audio. Typically the AR Content relates to an AR Target

AR Marker A digital object displayed on the screen that indicates the availability of AR Content about an AR Target.

AR Target Any entity of the real world such as POI, product, person, vehicle, etc. that can have associated AR Content.

AR View The view provided by AR application that supports user watching AR Content on the screen in overlay to the camera video stream.

User Context A set of dynamic information that describes the current general status of the user and his/her nearby environment.

This set of information can be retrieved from a variety of sources including OMA enablers

AR App The AR App is an external entity resident on the device that requests and receives AR Content(s) from MobAR Client, and presents them to the user. AR App also report AR Metrics data to the MobAR Client.

MobAR Client The MobAR Client is the device side functional component of the MobAR Enabler

Content Provider See [OMADICT]

MobAR Server The MobAR Server is the network side functional component of the MobAR Enabler

User See [OMADICT]

3.3 Abbreviations

OMA	Open Mobile Alliance
API	Application Programming Interface
AR	Augmented Reality
MobAR	Mobile Augmented Reality
POI	Point Of Interest

4. Introduction

Augmented Reality (AR) is a new interactive mechanism that allows to enrich the real world with virtual digital objects, visible only trough an electronic device (e.g. camera view stream of the mobile phone).

In this context, the ecosystem for the AR field envisages several actors and technological building blocks that interwork in order to build AR services.

The OMA Mobile Augmented Reality (OMA MobAR) aims at providing an overall framework to enable mobile AR services that can assure cross-platform exchange of AR Content and universal access to AR Content by providing new mechanisms for transport, filtering and personalization of AR Contents. OMA MobAR is built over a client-server model, with the MobAR Client that gets from the MobAR Server information and content that can "augment" the scene the user is looking at through his/her mobile device.

4.1 Version 1.0

The version 1.0 of the MobAR Enabler covers mainly the interaction between MobAR Client and MobAR Server through messages definition on MobAR-2 and MobAR-3 interfaces.

The main functionalities supported on MobAR-2 interface are:

- AR Target Search
- AR Content Retrieve
- AR Content Subscribe/Unsubscribe
- AR Interaction Report

The main functionality supported on MobAR-3 interface is:

- AR Content Push.

5. Interface Specifications

5.1 Commonly Used Parameters

The following table contains descriptions of commonly used parameters for MobAR interfaces:

Parameter name	Data type	Description
ARClientID	String	The unique identifier of the MobAR client, and its generation and assignment are out of scope. The value MAY be MSISDN, DPE ClientID, etc. **Used in**: ARContentRetrieveRequest, ARTargetSearchRequest, ARContentSubscribeRequest, ARInteractionReport
ARServerID	String	The unique identifier of the MobAR Server, and its generation and assignment are out of scope. **Used in**: ARContentRetrieveResponse, ARTargetSearchResponse, ARContentSubscribeResponse
Version	String	The version of the MobAR Enabler currently supported. **Used in**: ARContentRetrieveRequest, ARTargetSearchRequest, ARContentSubscribeRequest, ARInteractionReport, ARContentRetrieveResponse, ARTargetSearchResponse, ARContentSubscribeResponse

5.2 MobAR-1

The interface definition of MobAR-1 is postponed to a future release and is not in the scope of MobAR 1.0.

5.3 MobAR-2

The MobAR-2 interface allows the MobAR Client to:

- Request and search AR Target information from the MobAR Server
- Request and retrieve AR Content from the MobAR Server
- Subscribe or unsubscribe for AR Content from the MobAR Server
- Request the MobAR Server to set or update of user preferences
- Report collected AR Metrics data and User’s feedback to the MobAR Server

The MobAR-2 interface allows the MobAR Server to:

- Receive the AR Target search request from the MobAR Client and provide the AR Target search response to the MobAR Client
- Receive the AR Content retrieve request from the MobAR Client and provide the AR Content retrieve response to the MobAR Client
- Receive the subscribe or unsubscribe request for AR Content from the MobAR Client

- Receive the set or update of user preferences request from the MobAR Client
- Receive the AR Metrics data and User’s feedback report from the MobAR Client

5.3.1 AR Target Search

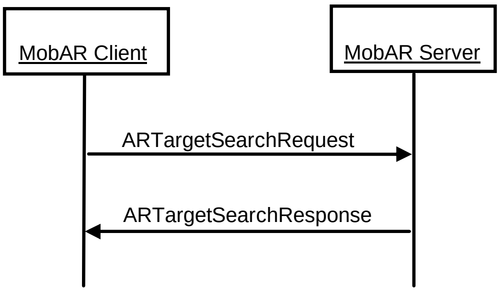

Figure 1: Flow of the AR Target Search Procedure

Messages included in the AR Target Search Procedure:

Message	Requirement	Direction
ARTargetSearchRequest	Mandatory	MobAR Client → MobAR Server
ARTargetSearchResponse	Mandatory	MobAR Client ← MobAR Server

5.3.1.1 ARTargetSearchRequest message

This message is used by the MobAR Client to request AR Target information from the MobAR Server.

In this request, the MobAR Client SHALL be able to pass the parameters as described in the first column of Table 1: ARTargetSearchRequest message parameters

.

Parameter name	XML Type	Data type	Cardinality	Description
ARTargetSearchRequest	E			Indicates the name of the message.
ARClientID	A	String	0…1	See description in section 5.1.
Version	A	String	1	See description in section 5.1.
SearchCondition	E1	SerachCondition Structure	1	The parameter used to search AR Targets based on different AR Target characteristics such as geolocation or real-time captured image feature
FiltrationCriteria	E1	FiltrationCriteria Structure	0...1	The parameter used to filter AR Contents according to user context and user preference
ViewCondition	E1	ViewCondition Structure	0..1	The parameter used to indicates the view conditions.

Table 1: ARTargetSearchRequest message parameters

5.3.1.2 ARTargetSearchResponse message

This message is used by the MobAR Server to return AR Target information response to the MobAR Client.

In this response, the MobAR Server SHALL be able to pass the parameters as described in the first column of Table 2.

Parameter name	XML Type	Data type	Cardinality	Description
ARTargetSearchResponse	E			Indicates the name of the message.
ARServerID	A	String	0…1	See description in section 5.1.
Version	A	String	1	See description in section 5.1.
ARTargetGroup	E1	ARTargetGroup Structure	0…N	It represents a group of AR Targets.
ResponseStatus	E1	Enumeration	0…1	It indicate the status of the reponse. The value is specified in the value column in section 7.3.4.

Table 2: ARTargetSearchResponse message parameters

5.3.2 AR Content Retrieve

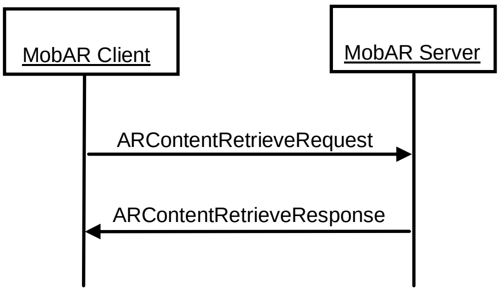

Figure 2: Flow of the AR Content Retrieve Procedure

Messages included in the AR Content Retrieve Procedure:

Message	Requirement	Direction
ARContentRetrieveRequest	Mandatory	MobAR Client → MobAR Server
ARContentRetrieveResponse	Mandatory	MobAR Client ← MobAR Server

5.3.2.1 ARContentRetrieveRequest message

This message is used by the MobAR Client to request AR Content from the MobAR Server.

In this request, the MobAR Client SHALL be able to pass the parameters as described in the first column of Table 3.

Parameter name	XML Type	Data type	Cardinality	Description
ARContentRetrieveRequest	E			Indicates the name of the message.
ARClientID	A	String	0…1	See description in section 5.1.
Version	A	String	1	See description in section 5.1.

ARTargetID	E1	String	1..N	The parameter used to identify the AR Target.
FiltrationCriteria	E1	FiltrationCriteria Structure	0..1	The auxiliary parameter used to filter AR Contents according to user context and user preference

Table 3: ARContentRetrieveRequest message parameters

5.3.2.2 ARContentRetrieveResponse message

This message is used by the MobAR Server to return AR Content response to the MobAR Client.

In this response, the MobAR Server SHALL be able to pass the parameters as described in the first column of Table 4.

Parameter name	XML Type	Data type	Cardinality	Description
ARContentRetrieveResponse	E			Indicates the name of the message.
ARServerID	A	String	0…1	See description in section 5.1.
Version	A	String	1	See description in section 5.1.
RequestID	E1	String	0..1	The parameter SHALL contain the SubscriptionID in the case of AR Content push to correlate the AR subscription session.
ARTarget	E1	ARTarget structure	1..N	The actual augmented information related to AR Target and/or AR Content.
ResponseStatus	E1	Enumeration	0…1	It indicates the status of the reponse. The value is specified in the value column in section 7.3.4.

Table 4: ARContentRetrieveResponse message parameters

5.3.3 AR Content Subscribe

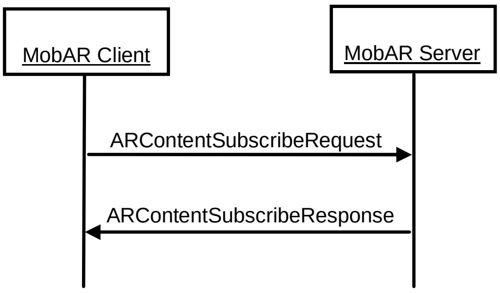

Figure 3: Flow of the AR Content Subscribe Procedure

Messages included in the AR Content Subscribe Procedure:

Message	Requirement	Direction

ARContentSubscribeRequest	Mandatory	MobAR Client → MobAR Server
ARContentSubscribeResponse	Mandatory	MobAR Client ← MobAR Server

5.3.3.1 ARContentSubscribeRequest message

This message is used by the MobAR Client to subscribe AR Content from the MobAR Server.

This message can also be used by the MobAR Client to request to unsubscribe AR Content to the MobAR Server.In this request, the MobAR Client SHALL be able to pass the parameters as described in the first column of Table 5.

Parameter name	XML Type	Data type	Cardinality	Description
ARContentSubscribeRequest	E			Indicates the name of the message.
ARClientID	A	String	0…1	See description in section 5.1.
Version	A	String	1	See description in section 5.1.
SubscriptionID	E1	String	0..1	The parameter used to identify AR Subscription session.
SubscriptionRule	E1	SubscriptionRule Structure	0..1	The parameter used to manage AR Contents push settings. It SHALL be included in the AR Content subscribe request, and May not be included in the AR Content unsubscribe request.
Duration	E1	duration	1	The duration of AR Subscription session. The lexical space of this data type is xsd:duration is the format defined by ISO 8601 under the form: “PnYnMnDTnHnMnS”.

Table 5: ARContentSubscribeRequest message parameters

5.3.3.2 ARContentSubscribeResponse message

This message is used by the MobAR Server to return AR Content subscription response to the MobAR Client.

This message can also be used by the MobAR Server to return AR Content unsubscribe response to the MobAR Client.In this response, the MobAR Server SHALL be able to pass the parameters as described in the first column of Table 6.

Parameter name	XML Type	Data type	Cardinality	Description
ARContentSubscribeResponse	E			Indicates the name of the message.
ARServerID	A	String	0…1	See description in section 5.1.
Version	A	String	1	See description in section 5.1.
SubscriptionID	E1	String	1	The parameter used to identify AR Subscription session.
SubscriptionStatus	E1	Enumeration	1	The parameter indicates the status of AR Subscription.
Duration	E1	duration	0…1	The duration of AR Subscription

				session. The lexical space of this data type is xsd:duration is the format defined by ISO 8601 under the form: “PnYnMnDTnHnMnS”.

Table 6: ARContentSubscribeResponse message parameters

5.3.4 AR Interaction Report

The AR Interaction Report operation from MobAR Client to MobAR Server is performed over the MobAR-2 and it is used to report the collection of user interactions and associated metrics related to the AR Target and AR Content consumption. This operation consists of one message:

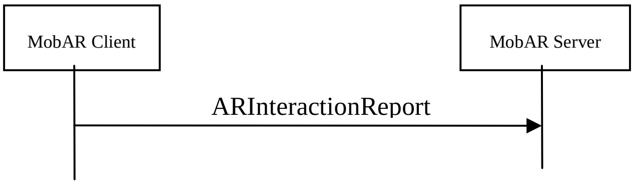

Figure 4: Flow for ARInteractionReport operation

Message	Requirement	Direction
ARInteractionReport	Mandatory	MobAR Client → MobAR Server

5.3.4.1 ARInteractionReport message

Name	XML Type	Data Type	Cardinality	Description
ARInteractionReport	E			Indicates the name of the message.
ARClientID	A	String	0…1	See description in section 5.1.
Version	A	String	1	See description in section 5.1.
ARTargetID	E1	String	1	Identify the specific AR Target on which the user provides his interaction
ARInteraction	E1	ARInteraction Structure	1	Contains the interaction provided or updated by the user on the AR Target and associated AR Content

Table 7: ARInteractionReport message

5.4 MobAR-3

The MobAR-3 interface allows the MobAR Client to:

- Receive push delivery of AR Target information from the MobAR Server
- Receive push delivery of AR Content(s) from the MobAR Server

The MobAR-3 interface allows the MobAR Server to:

- Push AR Target information to the MobAR Client
- Initiate AR Content push request to the MobAR Client

5.4.1 AR Content Push

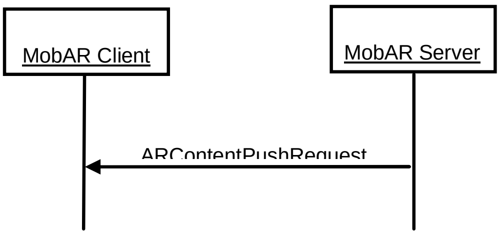

Figure 5: Flow of the AR Content Push Procedure

Message included in the AR Content Push Procedure:

Message	Requirement	Direction
ARContentPushRequest	Mandatory	MobAR Client ← MobAR Server

5.4.1.1 ARContentPushRequest message

This message is used by the MobAR Server to push AR Target information / AR Content(s) to the MobAR Client.

The ARContentPushRequest message SHALL be formatted as the ARContentRetrieveReponse message defined in the section 5.3.2.2 with the following clarifications:

- RequestID SHALL contain the value of SubscriptionID provided by the MobAR Server in the ARContentSubscriptionResponse message.
- ARServerID parameter SHALL be included.

5.5 MobAR-4

The interface definition of MobAR-4 is postponed to a future release and is not in the scope of MobAR 1.0.

6. Components Functionalities and Procedures

6.1 MobAR Client Operations and Functions

6.1.1 MobAR Client Settings Handling

The MobAR Client SHALL allow the User to select which information make available to the MobAR Server. In particular, in order to protect the user privacy, the following user information MAY NOT be provided:

- Location
- Category
- AR Interaction: metrics data and User's feedback

The MobAR Client SHALL verify if the usage of Location and Category are enabled by the User before sending them to the MobAR Server in messages where the SearchCondition parameter is needed.

The MobAR Client SHALL verify if the collection of the AR Interaction is enabled before sending the ARInteractionReport message to the MobAR Server.

The MobAR Client SHALL allow the User to locally set and update the following User settings and preferences:

- Radius for AR Target and AR Content search;
- Categories of a specific AR Target type (e.g. for a POI it may be restaurant, theater);
- SubscriptionRules

which can be used in subsequent requests (e.g. ARTargetSearch, …) if new or updated settings and preference are not provided by the User.

6.1.2 AR User Context Handling

The MobAR Client SHOULD determine the need of AR user context access according to the information source of the following message parameters, before constructing and sending them to the MobAR Server:

- The value of SearchCondition sturcture contained in the ARTargetSearchRequest messgage and ARContentSubscribeRequest messgage
- The value of FiltrationCriteria sturcture contained in the ARTargetSearchRequest messgage, ARContentRetrieveRequest messgage and ARContentSubscribeRequest messgage
- The value of TriggerCondition sturcture contained in the ARContentSubscribeRequest messgage

The MobAR Client can access AR user context according to the current status of sensors, other enabler clients and Device Capabilities on the user device and the availability of external network capabilities such as OMA Enablers.

Note: The underlying mechanism for actual user context access is out of the scope of MobAR Enabler.

6.1.3 AR Target Search

The MobAR Client can request AR Target search towards MobAR Server based on the reception of AR-related functionalities access request from AR App.

The MobAR Client SHALL send the ARTargetSearchRequest message defined in section 5.3.1.1 to the MobAR Server.

The MobAR Client SHOULD provide the SearchCondition parameter defined in section 7.2.1 in the ARTargetSearchRequest message to set search condition for the AR Target search.

The MobAR Client SHOULD obtain user geolocation information and provide Location element defined in section 7.2.6for the SearchCondition parameter.

Note: the underlying sources of location information can be Global Positioning System (GPS) and location inferred from network signals such as IP address, RFID, WiFi and GSM/CDMA cell IDs, as well as user input, which are out of the scope of MobAR Enabler.

If user preference or setting for search radius is available, the MobAR Client SHOULD provide SearchRadius element defined in section 7.2.1for the SearchCondition parameter to set AR Target search radius.

If current orientation of AR device is available, the MobAR Client SHOULD provide Orientation element defined in section 7.2.7for the SearchCondition parameter to set AR Target search direction.

If user preference or setting for relationship search is available, the MobAR Client SHOULD provide Relationship element defined in section 7.3.2for the SearchCondition parameter to indicate AR Target relationships and to request the search of AR Target which has such relationships.

If any multimedia content can be used for AR Target search is available, the MobAR Client SHOULD provide Multimedia element defined in section 7.2.3 for the SearchCondition parameter.

Note: in case of visual search, specific target feature extraction algorithms are out of the scope of MobAR Enabler.

The MobAR Client SHOULD provide the FiltrationCriteria parameter defined in section 7.2.2in the ARTargetSearchRequest message to set filtration criteria for the AR Target search.

The MobAR Client SHOULD provide the ViewCondition parameter defined in section 7.2.5 in the ARTargetSearchRequest message to set the preferred view of content retrieved.

The AR Target information is expected to be sent back from the MobAR Server in the ARTargetSearchResponse message defined in section 5.3.1.2. Upon receiving this message, the MobAR Client SHALL extract all the searched AR Target information and can perform some additional processing (e.g. more filtering based on local information available on the device, …) before make them available to AR Apps.

6.1.4 AR Content Retrieve

The MobAR Client can request AR Content retrieval towards MobAR Server based on the reception of AR Content access requests from AR App or user interaction such as AR Marker selection and click to view details.

The MobAR Client SHALL send the ARContentRetrieveRequest message defined in section 5.3.2.1 to the MobAR Server.

The MobAR Client SHOULD provide one or more ARTargetID parameters defined in section 5.3.2.1 in the ARContentRetrieveRequest message to identify AR Target and request the retrieval of associated AR Content.

The MobAR Client can provide the FiltrationCriteria parameter defined in section 7.2.2 in the ARContentRetrieveRequest message to set filtration criteria for the AR Content retrieval.

The AR Contents and possible AR Target information are expected to be delivered back from the MobAR Server with the ARContentRetrieveResponse message defined in section 5.3.2.2. Upon receiving this message, the MobAR Client SHALL extract all the retrieved AR Contents and possible AR Target information, and can perform some additional processing (e.g. more filtering based on local information available on the device, …) before make them available to AR App.

6.1.5 AR Content Subscribe

In order to subscribe AR Content, the MobAR Client SHALL send the ARContentSubscribeRequest message defined in section 5.3.3.1 to the MobAR Server.

The MobAR Client SHALL provide the SubscriptionID parameter defined in section 5.3.3.1 when available in the ARContentSubscribeRequest message to identify the AR Content subscription session in order to update or refresh it.

The MobAR Client SHALL provide the SubscriptionRule parameter defined in section 7.2.8 in the ARContentSubscribeRequest message to specify AR Content push settings for the initiation of AR Content push notification.

The MobAR Client SHALL provide the Duration parameter defined in section 5.3.3.1 in the ARContentSubscribeRequest message to set an expiration time of the AR Content subscription session.

If the corresponding AR Targets are available, the MobAR Client SHOULD provide the ARTargetID element defined in section 7.2.8for the SubscriptionRule parameter.

If the search conditions for AR Target/AR Content search are available, the MobAR Client SHOULD provide the SearchCondition element defined in section 7.2.1for the SubscriptionRule parameter.

If user preference related to AR Content push trigger is available, the MobAR Client SHOULD provide the TriggerCondition parameter defined in section 7.2.9 for the SubscriptionRule parameter. The TriggerCondition parameter can be provided with one or more of the following elements: PushInterval, Presence and Location.

If user context and/or user preference related to AR Content filtering is available, the MobAR Client SHOULD provide the FiltrationCriteria parameter defined in section 7.2.2for the SubscriptionRule parameter.

Upon receiving the ARContentSubscribeResponse message defined in section 5.3.3.2 from the MobAR Server, the MobAR Client SHALL extract the SubscriptionID parameter and the SubscriptionStatus parameter to identify the AR Content subscription session and its status. The MobAR Client SHALL get ready to receive AR Content push notifications initiated by the MobAR Server based on AR Content push settings and receive AR Content when the status is active.

The MobAR Client SHALL also extract the Duration parameter to confirm the duration of AR Content subscription session.

6.1.6 AR Content Unsubscribe

Once one specific AR Content subscription session has been established between the MobAR Client and the MobAR Server, the MobAR Client can request the MobAR Server to terminate the subscription of AR Content push notification.

The MobAR Client SHALL send the ARContentSubscribeRequest message defined in section 5.3.3.1 to the MobAR Server.

The MobAR Client SHALL provide the specific SubscriptionID parameter in the ARContentSubscribeRequest message, defined in section 5.3.3.1, to identify the AR Content subscription session needs to be terminated.

The MobAR Client SHALL provide the Duration parameter with the value zero in the ARContentSubscribeRequest message, defined in section 5.3.3.1, to indicate to unsubscribe the AR Content subscription.

Upon receiving the ARContentSubscribeResponse message defined in section 5.3.3.2 from the MobAR Server, the MobAR Client SHALL extract the SubscriptionID parameter and the Status parameter to identify the AR Content subscription session and its status to verify the unsubscribe request is successful.

6.1.7 AR Interaction Report

Based on the performed AR Interaction between AR user and available AR Targets, the MobAR Client SHALL collect the AR Interaction, create an AR Interaction Report request according to the ARInteractionReport message as defined in section 5.3.4.1and send it to the MobAR Server containing:

- The reference to the specific AR Target on which the user provides his interaction trough the ARTargetID parameter
- The indication of the interaction performed by the user over the AR Target through the ARInteraction structure as defined in section 7.2.13. Based on the action performed by the user the MobAR Client SHALL include one of the following values for the Interaction parameter:
    - “select” if user clicked on the AR Target or AR Content
    - “click on website” if the user clicked on the website related to the AR Target
    - “click to call” if the user clicked on the phone details to make a call
    - “click to mail” if the user clicked on the e-mail details to send an email

If the action corresponds to a vote or a comment, the MobAR Client SHALL include the related value on Vote or Comment parameter.

6.2 MobAR Server Operations and Functions

6.2.1 AR Target Search

Upon reception of the ARTargetSearchRequest message defined in section 5.3.1.1 from the MobAR Client, the MobAR Server SHALL extract all the parameters related to search, to identify which AR Targets need to be recognized and returned.

The MobAR Server SHALL perform AR Target search operation based on the SearchCondition parameter defined in section 7.2.1, which contains user preferences related to AR Target search condition.

The MobAR Server SHALL search AR Target based on current user geolocation, provided as the value of Location element.

When the SearchRadius element is contained in the SearchCondition parameter, the MobAR Server SHALL limit search range to the value of SearchRadius element during AR Target search operation.

When the Orientation element is contained in the SearchCondition parameter, the MobAR Server SHOULD limit search direction to the value of Orientation element during AR Target search operation.

When the Relationship element is contained in the SearchCondition parameter, the MobAR Server SHOULD determine AR Target relationships based on the value of Relationship element, and futher search for AR Target which has such relationships with those AR Targets already searched back.

When the Multimedia element is contained in the SearchCondition parameter, the MobAR Server SHOULD perform AR Target search operation according to the content type of Multimedia element value. For example, in the case of visual search, compact descriptor or image can be used for AR Target recognition and match.

The MobAR Server SHALL perform AR Target filtration operation based on the FiltrationCriteria parameter defined in section 7.2.2, which contains user preferences related to AR Target filtration criteria.

Based on the value of ViewCondition parameter defined in section 7.2.5, the MobAR Server SHALL retrieved the appropriate AR Target/AR Content and associated information. It SHOULD also perform AR Target/AR Content aggregation functionality based on the value of ViewType defined in section 7.3.1:

- if ViewType is “map”, then the grouping SHALL be performed based on shared location among AR Targets considering the MinGroupRadius value as radius of the area where realize the grouping of AR Targets in AR Target Group.
- if ViewType is “ar”, then the grouping SHALL be performed based on shared direction among AR Targets considering the NumAngularSections value as the number of angular sections the 360° perspective will be divided in and where to compute if the AR Target belong to the related ARTargetGroup.

The MobAR Server SHALL send all the searched AR Target information and possible cached AR Contents, back to the MobAR Client, via the ARTargetSearchResponse message defined in section 5.3.1.2.

6.2.2 AR Content Retrieve

Upon reception of the ARContentRetrieveRequest message defined in section 5.3.2.1 from the MobAR Client, the MobAR Server SHALL extract all the parameters related to AR Content retrieval, to identify which AR Contents and AR Target information need to be retrieved and returned.

The MobAR Server SHALL perform AR Content retrieval operation based on the ARTargetID parameter defined in section 5.3.2.1, which contains unique identifier of AR Targets related to AR Content.

The MobAR Server SHOULD retrieve AR Content based on the available AR Content management metadata, such as availability, visiting address (Web URL) and deployment rules.

The MobAR Server SHALL perform AR Content filtration operation based on the FiltrationCriteria parameter defined in section 7.2.2, which contains user preferences related to AR Content filtration criteria.

The MobAR Server SHALL send all the retrieved AR Contents and ARTargetback to the MobAR Client, via the ARContentRetrieveResponse message defined in section 5.3.2.2.

6.2.3 AR Content Subscribe

Upon reception of the ARContentSubscribeRequest message defined in section 5.3.3.1 from the MobAR Client, the MobAR Server SHALL extract all the parameters related to AR Content subscription, to maintain the AR Content subscription session and manage AR Content push settings for the initiation of AR Content push notification.

The MobAR Server SHALL check if the SubscriptionID parameter is available in the ARContentSubscribeRequest message. If it is not present, the MobAR Server SHALL assign an unique identifer as the value of SubscriptionID. If it is present, the MobAR Server SHALL identify the AR subscription session according to the value of SubscriptionID parameter.

The MobAR Server SHALL check the Duration parameter in the ARContentSubscribeRequest message, and maintain the lifecycle of AR Content subscription session based on its value and service provider policy. If the Duration parameter has the value zero, the MobAR Server SHALL terminate the AR Content subscription session and delete AR Content push settings according to the operations and functions described in the section 6.2.5.

The MobAR Server SHALL check the SubscriptionRule parameter in the ARContentSubscribeRequest message, and use it in order to manage the subscription session (e.g. create or update AR Content push settings for the initiation of AR Content push notification).

If available, the MobAR Server can access and use AR user context information in order to determine whether to initiate the AR Content push notification based on the value of TriggerCondition element of SubscriptionRule parameter.

If the ARTargetID element is available in the SubscriptionRule parameter, the MobAR Server SHOULD access the status of subscribed AR Contents according to the value of ARTargetID element and determine whether to initiate the AR Content push notification based on the publish status, during the lifecycle of AR Content subscription session.

If the PushInterval element is available in the TriggerCondition parameter, the MobAR Server SHOULD calculate the time interval according to the value of PushInterval element and determine whether to initiate the AR Content push notification based on calculated time interval, during the lifecycle of AR Content subscription session.

If available, the MobAR Server can subscribe and use AR user presence information to be matched against the Presence element of TriggerCondition parameter (if sent), to determine whether to initiate the AR Content push notification based on current AR user presence during the lifecycle of AR Subscription session.

If available, the MobAR Server can access and use AR device location information to be matched against the Location element of TriggerCondition parameter (if sent), to determine whether to initiate the AR Content push notification based on accessed AR device location during the lifecycle of AR Content subscription session.

Note: in the process of evaluating whether to initiate AR Content push notification, different trigger conditions should remain independent with each other.

Once one certain trigger condition is met, the MobAR Server SHALL retrieve AR Contents based on the value of SearchCondition element and FiltrationCriteria element of SubscriptionRule parameter and initiate the AR Content push request towards the MobAR Client.

Via the ARContentSubscribeResponse message defined in section 5.3.3.2, the MobAR Server SHALL return AR Content subscription status back to the MobAR Client.

The MobAR Server SHALL provide the SubscriptionID parameter in the ARContentSubscribeResponse message, to identify the AR Content subscription session.

The MobAR Server SHALL provide the Status parameter in the ARContentSubscribeResponse message to indicate the status of AR Content subscription session. The possible value of the Status parameter can be “active”, “pending” or “terminated”.

The MobAR Server SHALL provide the Duration parameter in the ARContentSubscribeResponse message to set the duration of AR Content subscription session.

Note: the MobAR Server MUST NOT provide a value for the Duration parameter to lengthen the subscription.

6.2.4 AR Content Unsubscribe

Upon reception of the ARContentSubscribeRequest message from the MobAR Client, in which the Duration parameter with the value zero has been found, the MobAR Server SHALL extract all the parameters related to AR Content unsubscription, to terminate the AR Content subscription session.

The MobAR Server SHALL check the SubscriptionID parameter in the ARContentSubscribeRequest message, to identify the AR Content subscription session need to be terminated.

The MobAR Server SHALL check the Duration parameter in the ARContentSubscribeRequest message, to confirm the intention to unsubscribe the AR Content subscription session. The MobAR Server SHALL terminate the maintenance of AR Content subscription session.

Via the ARContentSubscribeResponse message defined in section 5.3.3.2, the MobAR Server SHALL send AR Content unsubscription status back to the MobAR Client.

The MobAR Server SHALL provide the same SubscriptionID parameter in the ARContentSubscribeResponse message, as that of received unsubscribe request to identify AR Content subscription session.

The MobAR Server SHALL provide the Status parameter with value “terminated” in the ARContentSubscribeResponse message to indicate the termination status of AR Content subscription session.

6.2.5 AR Interaction Report

After receiving the ARInteractionReport message from the MobAR Client, the MobAR Server SHALL store the ARInteraction received and associate it to user and related ARTarget or AR Content in order to use it in the future AR Content selection process.

6.2.6 Device Capabilities Access

After receiving the ARContentRetrieveRequest, ARTargetSearchRequest and ARContentSubscribeRequest from MobAR Client, the MobAR Server MAY retrieve device capabilities from the DPE Server using the DPE-3 interface.

In this case the MobAR Server acts as Service Provider and can query the value of a single property or multiple properties of the device using the DPE-3 Property-Query message as defined in [OMA-DPE-TS].

For this request it is mandatory to send the Client-Id that the MobAR Server SHALL retrieve using the DPE-3 Client-Id-Query message as defined in [OMA-DPE-TS].

After receiving the DPE-3 Property-Response message, the MobAR Server SHALL use the value of the requested device property(ies) to accordingly filter or personalize AR Target/AR Content to be selected.

7. Data Model

This section defines the data model that is used by MobAR interfaces.

7.1 XML Namespaces

The namespace for the MobAR data types is: “urn:oma:xml:mobar:1.0”.

7.2 Common Structures

7.2.1 SearchCondition Structure

Name	XML Type	Data Type	Cardinality	Description
Location	E1	Location Structure	0…1	The current location of AR user.
SearchRadius	E1	Float	0…1	The preset AR Target search radius. The unit of this parameter SHALL be meter.
Category	E1	String	0…N	The searched category for AR Targets.
Orientation	E1	Orientation Structure	0…1	The current orientation of AR device.
Relationship	E1	Enumeration	0…1	The 1-to-1 or 1-to-many relationships between AR Targets
Multimedia	E1	Multimedia Structure	0…1	Any multimedia content can be used for AR Target search such as compact descriptor, image, video, audio etc.

Table 8: SearchCondition Structure

7.2.2 FiltrationCriteria Structure

Name	XML Type	Data Type	Cardinality	Description
TargetFiltration	E1	ARTargetInfo Structure	0…N	The preference or service setting of AR user used to filter AR Targets e.g. name, description, provider of AR Target..
ContentFiltration	E1	String	0…N	The preference or service settings of AR user used to filter AR Contents e.g. media type of AR Content.
ServiceFiltration	E1	String	0…1	It indicates the specific service for which the AR Targets and AR Content are requested e.g. for touristic purpose, for advertising, …

Table 9: FiltrationCriteria Structure

7.2.3 Multimedia Structure

Name	XML Type	Data Type	Cardinality	Description
ID	A	String	1	A unique identifier of multimedia content provided.
Content	E1	Content Structure	0…1	It represents the actual content. At least one between Content or ContentUrl MUST be included.

Table 10: Multimedia Structure

7.2.4 Content Structure

Name	XML Type	Data Type	Cardinality	Description
ContentType	A	String	1	It Indicates the type of the multimedia content provided
ContentUrl	A	URI	0…1	It represents a reference to content. At least one between Content or ContentUrl MUST be included.
ActualContent	E1	Binary	0…1	It represents the actual content. At least one between Content or ContentUrl MUST be included.

Table 11: Content Structure

7.2.5 ViewCondition Structure

Name	XML Type	Data Type	Cardinality	Description
ViewType	A	ViewType Enumeration	1	Indicates the AR Content consumption style. The default value is “list”.
MinGroupRadius	E1	Float	0…1	Indicates the minimum radius to be considered of the area where realize the grouping of AR Targets in AR Target Group. It SHALL be included if viewType is “map”. It SHALL NOT be included otherwise.
NumAngularSections	E1	Integer	0…1	Indicates the number of angular sections the 360° perspective will be divided in. Each angular section will be (360/NumAngularSections)°. Each angular section SHALL contain no more than one AR Target Group. It SHALL be included

				if viewType is “ar”. It SHALL NOT be included otherwise. NumAngularSections SHALL not be greater than 60.

Table 12: ViewCondition Structure

7.2.6 Location Structure

Name	XML Type	Data Type	Cardinality	Description
Latitude	E1	Float	1	Latitude WGS84 Decimal Degrees (DD) format: e.g. “38.889722” Latitude values are bounded by ±90°.
Longitude	E1	Float	1	Longitude WGS84 Decimal Degrees (DD) format: e.g. “-47.945267” Longitue values are bounded by ±180°
UncertaintyRadius	E1	Float	0…1	Uncertainty radius in meters. It refers to Location expressed as Latitude and Longitude. If Latitude and Longitude are not present then it MUST NOT be included.
Address	E1	String	0…1	It represents a civic address e.g. mailing address, street address.
City	E1	String	0…1	It indicates the city of the AR Target
PostalCode	E1	String	0…1	It indicates the postal code of the AR Target.
Country	E1	String	0…1	It indicates the country of the AR Target.

Table 13: Location Structure

7.2.7 Orientation Structure

Name	XML Type	Data Type	Cardinality	Description
Heading	E1	Double	1	The clockwise rotation angle around z (Up) axis.
Tilt	E1	Double	1	The clockwise rotation angle around x (East) axis.
Roll	E1	Double	1	The clockwise rotation angle around y

				(North) axis.

Table 14: Orientation Structure

7.2.8 SubscriptionRule Structure

Name	XML Type	Data Type	Cardinality	Description
ARTargetID	E1	String	0...N	The parameter used to determine AR Contents need to be pushed.
SearchCondition	E1	SearchCondition Structure	0…1	The parameter used to find AR Contents need to be pushed.
TriggerCondition	E1	TriggerCondition Structure	0…1	The parameter used to trigger AR Content push notifications.
FiltrationCriteria	E1	FiltrationCriteria Structure	0…1	The auxiliary parameter used to filter AR Contents according to user context and user preference.

Table 15: SubscriptionRule Structure

7.2.9 TriggerCondition Structure

Name	XML Type	Data Type	Cardinality	Description
PushInterval	E1	duration	0…1	The time interval condition for AR Content push. The lexical space of this data type is xsd:duration is the format defined by ISO 8601 under the form: “PnYnMnDTnHnMnS”.
Presence	E1	String	0…1	The user persence condition for AR Contents push.
Location	E1	Location Structure	0…1	The device location condition for AR Contents push.

Table 16: TriggerCondition Structure

7.2.10 ARTargetGroup Structure

Name	XML Type	Data Type	Cardinality	Description
AngularSectionIndex	A	Integer	0…1	It represents the angular section index of the current ARTargetGroup, starting from the North (AngularSectionIndex=“0”) and going clockwise. It SHALL be included if the viewType is ”ar”. It SHALL NOT be included otherwise.
ARTargetGroupID	A	String	1	It represents the identification of an AR

				Cluster
ARTargetGroupLocation	E1	Location Structure	0…1	It represents the location where the ARTargetGroup is localized in the map view. It indicates the location of the AR Marker that represent the group. It SHALL be included if the viewType is ”map”. It SHALL NOT be included otherwise.
ARTarget	E1	ARTarget Structure	1…N	The parameter used to identify the list of AR Target. The parameter is used to contain AR Target information and/or associated AR Content.

Table 17: ARTargetGroup Structure

7.2.11 ARTarget Structure

Name	XML Type	Data Type	Cardinality	Description
ARTargetID	A	String	1	The parameter used to identify the AR Target.
ARTargetInfo	E1	ARTargetInfo Structure	0…1	The parameter used to represent AR Target information
ARContent	E1	Multimedia Structure	0…N	The parameter used to contain actual augmented information related to AR Target

Table 18: ARTarget Structure

7.2.12 ARTargetInfo Structure

Name	XML Type	Data Type	Cardinality	Description
Position	E1	Location	1	It indicates the location information of the AR Target
Name	E1	String	1	It indicates the name of the AR Target.
Category	E1	String	0…1	It indicates the category of the AR Target.
WebsiteUrl	E1	URI	0…N	It indicates the web site URL or any web resource related to the AR Target.
Phone	E1	URI	0…N	It indicates the phone number belonging to the AR Target.
Email	E1	URI	0…N	It indicates an email address of the AR Target.

RatingAverage	E1	Float	0…1	It indicates the value of the average rating associated to the AR Target
NumRatings	E1	Integer	0…1	It indicates the total number of rating available for that AR Target
Provider	E1	String	0…1	It indicates the provider of AR Target.
Description	E1	String	0…1	It contains a general description of the AR Targeta and some more useful information about the AR Target.
Time	E1	String	0…1	It indicates the fixed time or sequence of times applicable to the AR Target.

Table 19: ARTargetInfo Structure

7.2.13 AR Interaction Structure

Name	XML Type	Data Type	Cardinality	Description
Interaction	E1	String	0…1	Specify the user interaction with displayed AR Target/AR Content. Some reserved strings are defined as value for this parameter as follows: • “select”: user clicked on the AR Target or AR Content • “click on website”: user clicked on the website related to the AR Target • “click to call” user clicked on the phone details to make a call • “click to mail” user clicked on the e-mail details to send an email
Vote	E1	Integer	0…1	Contain the actual value of the vote expressed by the user about the AR Target/AR Content. The value SHALL be in the range from 0 to 100.
Comment	E1	String	0…1	Contain a comment left by the user about the AR Target/AR Content.
Timestamp	E1	dateTime	1	Indicate the time where the AR Interaction was done.

Table 20: User Interaction structure

7.3 Enumerations

7.3.1 ViewType Enumeration

Enumeration	Description
List	It indicates that the view type is set to list.
Map	It indicates that the view type is set to map.
Ar	It indicates that the view type is set to ar.

Table 21: ViewType enumeration

7.3.2 Relationship Enumeration

Enumeration	Description
Parent	The parent relationship.
Child	The child relationship.
Historic	The relationship at a previous point in time.
Future	The relationship at a later point in time.
Contains	The geo-spatial contains relationship.
Within	The geo-spatial contained-within relationship.
Adjacent-to	The geo-spatial adjacent-to relationship such as intersects, crosses, touches.

Table 22: Relationship enumeration

7.3.3 SubscriptionStatus Enumeration

Enumeration	Description
Active	It indicates that the subscription has been accepted.
Pending	It indicates that the subscription has been received, but there is insufficient information to accept or deny the subscription at this time.
Terminated	It indicates that the subscription is not active for some reasons e.g. rejected or timeout.

Table 23: SubscriptionStatus Enumeration

7.3.4 ResponseStatus Enumeration Values

Value	Enumeration	Description
1	AR Target not found	It indicates that the requested AR Target is not available
2	Subscription ID not found	It indicates that the subscription ID was not found

3	Missing parameter	It indicates that a parameter is missing in the request.
4	Invalid parameter	It indicates that a parameter is not valid /allowed in the request.

Table 24: ResponseStatus Enumeration

8. Protocol Bindings

8.1 HTTP Bindings

8.1.1 General

MobAR Client and MobAR Server SHALL support Hypertext Transfer Protocol version 1.1 (HTTP1.1 [RFC2616]) for MobAR-2 interface.

MobAR Client and MobAR Server MUST support at least one of XML and JSON content types.

8.1.1.1 Media Type

MobAR Client SHALL support messages formatted as entity-bodies with the following media types:

- application/json or application/xml media type is used on MobAR-2 interface when sending the HTTP requests and when receiving the HTTP responses.

MobAR Server SHALL support messages formatted as entity-bodies with the following media types:

- application/json or application/xml media type is used on MobAR-2 interface when sending the HTTP responses and when receiving the HTTP request;

8.1.1.2 HTTP Method

All the request messages SHALL be send as HTTP POST method requests.

The following optional Headers may be included in the request messages.

- the receiver's address in the request line
- the Host request-header set to the hostname or IP address of the receiver.
- the User-Agent request-header set to identify the host device (e.g. "vendor-model/version"), and the name and version of the sender as user agent initiating the request.
- the Accept request-header with value "application/xml" or "application/json" as applicable
- the Accept-Encoding request-header with value per the supported HTTP compression encodings, i.e. deflate and / or gzip
- the Accept-Language request-header with value per the supported HTTP supported languages (e.g. en, *)
- the Accept-MsgSize is the maximum message size that terminal can handle.
- the Content-Length entity-header set to the length of the entity-body
- the Content-Type entity-header with value "application/xml", or "application/json"
- the MobAR-2 message(s) as message-body

If any of these headers are not present in the response to the request, the receiver SHALL assume their *default* values.

All the response messages SHALL be sending as response to the corresponding receiver's request as specified by the HTTP 1.1 including:

- Status-Line header reflects the outcome of the HTTP POST request
- the ETag entity-header set to a unique value within the scope of the receiver.

- the Content-Encoding entity-header set to the type of HTTP compression applied, if any
- the Content-Length entity-header set to the length of the entity-body
- the Content-Type entity-header with value “application/xml or “application/json”, as applicable
- the MobAR-2 message(s) as message-body, if the transaction is successful

8.1.1.3 HTTP Authentication

The MobAR Client and MobAR Server SHALL support HTTP Digest Authentication mechanisms (HTTP1.1 [RFC2617]) on MobAR-2 interface.

8.1.2 MobAR-2 Interface Message Bindings

The table below gives an overview of how MobAR-2 interface messages are bound to the HTTP based protocol stack.

For each request message MobAR Client MUST add a “message” parameter in the body of the HTTP POST request containing as value the message name as specified in the “Message” column of the following table.

The MobAR-2 interface messages MAY include HTTP Digest Authentication information.

When sending requests on MobAR-2 interface, the MobAR Client MAY insert user authentication and authorization information. If challenged by the MobAR Server, it SHALL insert user authentication and authorization information in the request.

When receiving requests on MobAR-2 interface, the MobAR Server MAY require user authentication and authorization information and challenge the MobAR Client.

Message	MobAR Client ←→ MobAR Server	HTTP Method
ARContentRetrieveRequest	→	HTTP POST
ARContentRetrieveResponse	←	HTTP Response (including 200 OK of the underlying method)
ARTargetSearchRequest	→	HTTP POST
ARTargetSearchResponse	←	HTTP Response (including 200 OK of the underlying method)
ARContentSubscribeRequest	→	HTTP POST
ARContentSubscribeResponse	←	HTTP Response (including 200 OK of the underlying method)
ARInteractionReport	→	HTTP POST

Table 25: MobAR-2 Interface messages bindings when uses HTTP protocol

8.1.3 JSON Encoding

8.1.3.1 Serialization rules: general conversion

Following guidelines provided in [REST_NetAPI_Common], general rules are provided for mapping between XML and JSON data formats:

a. XML elements that appear at the same XML hierarchical level (i.e. either root elements or within the same XML parent element), are mapped to a set of *name:value* pairs within a JSON object, as follows:

   (i) Each XML element appearing only once at the same hierarchical level ("single element") is mapped to an individual name:value pair. The name is formed according to bullet b, while the value is formed according to bullet c.

   (ii) XML elements appearing more than once at the same hierarchical level ("element list") are mapped to only one, individual name:value pair. The name is formed according to bullet b, while the value is a JSON array containing one value per each occurrence of the XML element. The name is formed according to bullet b whilst values are formed according to bullet c.

   (iii) Name and Value of JSON objects will go between "". Additionally, any JSON representation of an element of complex type will go between {}, according to [RFC4627].

b. The name of the name:value pair is the name of the XML elements (i.e. XML_element_name:value)

c. The value is formed as follows:

   (iv) when the XML element has neither attributes nor child XML elements, the value is equal to the value of the XML element. In case the element is nill (i.e it has no value), it will be indicated as having a "null" value within JSON.

   (v) when the XML element has child elements and/or attributes, the value is a JSON object containing the following name:value pairs:

   - one *name:value* pair per each attribute, where *name* is the name of the attribute and *value* is the value of the attribute.

   - one *name:value* pair associated to the text value (simple type content) of the XML element, where *name* is the string "$t" and *value* is the value of the XML element.

   - *name:value* pairs associated to XML child elements. These *name:value* pairs are formed in accordance with bullet a.

Within JSON, there is no need to reflect:

- the first <?xml version="1.0" encoding="UTF-8" ?> tag
- declaration of namespaces or schemaLocations

In order to generate unambiguous JSON from XML instances, based on the rules defined above, the following limitations need to be imposed on the XML data structures:

- it is not allowed that two different elements from different namespaces have the same name, in case they appear at the same level
- within an XML parent element, no attribute is allowed to have the same name as a child element of this parent element.

8.1.3.1.1 Examples (Informative)

The following is an example illustrating the guidelines:

Input XML content:

```
<Animals>
    <dog>
        <name attr="1234">Rufus</name>
        <Breed>labrador</Breed>
    </dog>
    <dog>
        <name>Marty</name>
        <Breed>whippet</Breed>
        <a/>
    </dog>
    <dog/>
    <cat name="Matilda"/>
    <a/>
</Animals>
```

Transformed JSON:

```
{"Animals": {
   "a": null,
   "cat": {"name": "Matilda"},
   "“dog": [
      {
         "Breed": "labrador",
         "name": {
            "$t": "Rufus",
            "attr": "1234"
         }
      },
      {
```

```
      "Breed": "whippet",

      "a": null,

      "name": "Marty"

    },

    null

  ]

}}
```

8.1.3.2 Rules for JSON-consuming applications

Applications that consume a JSON representation SHALL accept the following two different JSON representations for an array that contains one element:

- a pair of name and value (e.g. “name”: “one”)
- a pair of name and array of one value (e.g. “name”: [“one”])

Note: In JSON, according to [RFC4627], the order of objects is not significant, whilst the order of values within an array is.

8.2 OMA Push Bindings

The MobAR Server SHALL support either the Push Access Protocol (PAP) [PUSH-PAP] or the Push-OTA protocol [PUSH-OTA] for point-to-point delivery of the MobAR messages over the MobAR-3 interface.The MobAR Client SHALL support direct and indirect Push delivery of MobAR messages over MobAR-3 interface:

- MobAR-3 ARContentPush
- MobAR-3 ARTargetSearchResponse
- MobAR-3 ARContentRetrieveResponse

Whether to use PAP or Push-OTA is a deployment choice. Among other advantages, use of PAP prevents the MobAR Server from needing to implement most of the basic functions of OMA Push Proxy Gateway (PPG), e.g. the various Push-OTA protocols and target client context/capabilities awareness (as needed to select the appropriate transport bearer and protocol). Conversely, for high-volume search services limited to Push delivery over a single specific bearer (e.g. WAP1 Push over SMS), use of Push-OTA is fairly simple and avoids dependency upon a Push Proxy Gateway (PPG).

MobAR messages may not be directly deliverable over connectionless bearers such as WAP Push over SMS (limited to about 512 bytes) or SIP Push via the SIP MESSAGE method (limited to about 1300 bytes), as compared to WAP Push over HTTP or SIP Push via the INVITE/MSRP method (both of which support essentially unlimited content size). The alternative to direct delivery is “indirect delivery”, which involves the delivery of a Push notification message carrying a content location URL, from which the client retrieves the response.

When using PAP for direct MobAR messages delivery, the MobAR Server SHALL submit the MobAR messages using the MIME content type “application/xml” or “application/json”, as applicable, and MAY support various target client address schemes, e.g. PLMN, USER, SIP URI, IP address, etc.

When directly delivering MobAR messages via Push-OTA, the MobAR Server SHALL send the MobAR messages (headers and message body) encapsulated into a message/vnd.oma.push media type as described in [PUSH-MSG].

To deliver MobAR messages indirectly, the MobAR Server SHALL use the ServiceLoading [PUSH-SL] content type, via either PAP or Push-OTA as applicable, and include a URL from which the MobAR-Client can retrieve the actual MobAR message. MobAR Clients SHALL support indirect delivery of MobAR message, triggered by reception of ServiceLoading notifications.

For MobAR message, the MobAR Server SHALL include the Push Application ID header "X-Wap-Application-Id: x-oma-application:mobar.ua". Push Clients in MobAR supporting terminals SHALL support routing of Push messages with the Push Application ID header "X-Wap-Application-Id: x-oma-application:mobar.ua" to the MobAR Client. If there is no Push Client in the device, the MobAR Client SHALL implement the necessary Push Client functions for the supported Push-OTA protocol variants per [PUSH-OTA].

8.2.1 MobAR-3 Interface Message Bindings

The table below gives an overview of how MobAR-3 interface messages are bound to the OMA Push-OTA and PAP protocol.

Message	MobAR Client ←→ MobAR Server	HTTP Method	Push-OTA Binding	PAP Binding
MobAR-3: ARContentPush	←		Push Message (ARContentPush)	Push Message (ARContentPush)
		OR		
MobAR-3: ARContentPush Notification	←		ServiceLoading (ARContentPush URL)	Push Message (ServiceLoading (ARContentPush URL))
	→	HTTP GET		
MobAR-3: ARContentPush	←	HTTP Response (including 200 OK of the underlying method)		

Table 26: MobAR-3 ARContentPush Operation Bindings to OMA Push

Message	MobAR Client ←→ MobAR Server	HTTP Method	Push-OTA Binding	PAP Binding
MobAR-2: ARTargetSearchRequest	→	HTTP POST		
MobAR-3: ARTargetSearchResponse	←		Push Message (ARTargetSearchResponse)	Push Message (ARTargetSearchResponse)
		OR		
MobAR-3:	←		ServiceLoading	Push Message

Message	MobAR Client ←→ MobAR Server	HTTP Method	Push-OTA Binding	PAP Binding
ARTargetSearchResponse Notification			(ARTargetSearchResponse URL)	(ServiceLoading (ARTargetSearchResponse URL))
	→	HTTP GET		
MobAR-3: ARTargetSearchResponse	←	HTTP Response (including 200 OK of the underlying method)		

Table 27: MobAR-3 ARTargetSearch Operation Bindings to OMA Push

Message	MobAR Client ←→ MobAR Server	HTTP Method	Push-OTA Binding	PAP Binding
MobAR-2: ARContentRetrieveRequest	→	**HTTP POST**		
MobAR-3: ARContentRetrieveResponse	←		Push Message (ARContentRetrieveResponse)	Push Message (ARContentRetrieveResponse)
OR				
MobAR-3: ARContentRetrieveResponse Notification	←		ServiceLoading (ARContentRetrieveResponse URL)	Push Message (ServiceLoading (ARContentRetrieveResponse URL))
	→	HTTP GET		
MobAR-3: ARContentRetrieveResponse	←	HTTP Response (including 200 OK of the underlying method)		

Table 28: MobAR-3 ARContentRetrieve Operation Bindings to OMA Push

Note that the PAP Binding only applies to the MobAR Server.

9. Security considerations

MobAR Clients and MobAR Servers SHALL support HTTP over Transport Layer Security (TLS) as specified in [OMA TLS] over MobAR-2 interface in order to establish a secure communication.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version –or- No previous version within OMA

A.2 Draft/Candidate Version <current version> History

Document Identifier	Date	Sections	Description
Draft Versions OMA-TS-MobAR-V1_0-	01 Dec 2011	All	Establish the initial empty Technical Specification. Incorporated 53R01 and 54R01.
	17 Feb 2012	All	Applied the 2012 template, incorporated 58R01 and 59R01
	05 Mar 2012	5, 7, appendix C	Incorporated 62 (2011), 02R01 and 03R02
	23 Mar 2012	5	Incorporated 008R02
	02 Apr 2012	6	Incorporated 009R03
	26 Apr 2012	all	Incorporated 0010R01, 0011R02, 0012R02, 0013R01, 0014R02, 0015R01, 0016R01, 0017R02 and 0020R02.
	04 Jun 2012	7	Incorporated 0024R01
	12 Jun 2012	7	Incorporated 0023R02 and some editorial changes (section numbering formatting)
	04 Jul 2012	5, 7	Incorporated 19R02, 21 R01, 26 R01, 27 R01 and 28R01.
	16 Jul 2012	all	Incorporated 28R01 (partially incorporated in the previous version), 29R02, 31R01, 32R01, 33R01, 34R01, 35, 36R01, 37, 38R01 and 39R01.
	25 Jul 2012	all	Incorporated 30R05 and 42R01. Fixed some editorial issue (numebering of sections, table format, …).
	27 Jul 2012	A.2	It was corrected the mistake made on this table with respect to the implementation of agreed CRs
	26 Sep 2012	all	Incorporated: 48, 43R02, 45R01, 46, 51R01, 53R01, 54, 55R01, 56R01, 57R01, 59R01, 60R01 and 62. Resolved AI MobAR-2012-A004. Implemented changes based on OMA-CONRR-MobAR-V1_0-20120920-D
Candidate Version OMA-TS-MobAR-V1_0	16 Oct 2012	All	Status changed to Candidate by TP: OMA-TP-2012-0373-INP_MobAR_V1.0_ERP_and_ETR_for_Candidate_approval

Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for MobAR Client

Item	Function	Reference	Requirement
MobAR-C-001-O	Support for the XML request & response format	5, 7	
MobAR-C-002-O	Support for the JSON request & response format	5, 7	
MobAR-C-003-M	Support sending AR target search request and receiving response	5.3.1	
MobAR-C-004-M	Support sending AR content retrieval request and receiving response	5.3.2	
MobAR-C-005-M	Support sending AR content subscribe request and receiving response	5.3.3	
MobAR-C-006-M	Support sending AR content unsubscribe request and receiving response	5.3.3	MobAR-C-005-M
MobAR-S-007-M	Support receiving AR content push	5.4.1	MobAR-C-005-M
MobAR-C-008-M	Support sending AR interaction report	5.3.4	

B.2 SCR for MobAR Server

Item	Function	Reference	Requirement
MobAR-S-001-O	Support for the XML request & response format	5, 7	
MobAR-S-002-O	Support for the JSON request & response format	5, 7	
MobAR-S-003-M	Support receiving AR target search request and sending response	5.3.1	
MobAR-S-004-M	Support receiving AR content retrieval request and sending response	5.3.2	
MobAR-S-005-M	Support receiving AR content subscribe request and sending response	5.3.3	

Item	Function	Reference	Requirement
MobAR-S-006-M	Support receiving AR content unsubscribe request and sending response	5.3.3	MobAR-S-005-M
MobAR-S-007-M	Support sending AR content push	5.4.1	MobAR-S-005-M
MobAR-S-008-M	Support receiving AR interaction report	5.3.4	
MobAR-S-009-O	Support accessing to device capabilities	6.2.6	

Appendix C. Gap Analysis Considerations

This section provides the gap analysis with respect to the features related to Mobile Augmented Reality Enabler by identifying the available technologies and initiatives, including also the relevant OMA-enablers along with the possible recommendations for the reuse of them.

The following macro-areas will be covered as part of the scope of the MobAR Enabler:

- AR Content Data format
- AR Transport and Interfaces
- AR Client features
- Security & Privacy

C.1 AR Content Data Format

MobAR Needs	Identified Gaps	Relevant SDOs	Developing Emphasis	Specification
AR Markup Language				
AR Markup definition to support different AR Target(POI) sensing technologies: based on geolocation, based on image, natural feature or marker, based on end-user characteristics • AR Markup definition to support presentation of extensible AR Target (POI) information: location, category, type, time, name, description, relationships, URLs, contact information, multimedia content • Support for naming, indexing, and retrieving AR Content • Support for extensible AR Content format: 2D, 3D model, animation • Support for alternative AR Content consumption styles: AR View, list view, bird's eye view • Support for AR Content aggregation based on	• Lack of visual AR related tag support: marker, marker-less •Lack constructs support for rich augmentation: panoramic imagery, property animations •Lack capability of supplementary presentation: consumption styles, content aggregation •Lack capability of flexible interaction: event triggers and custom specific action	OGC	Geography Markup Language is an XML grammar written in XML Schema for the description of application schemas as well as the transport and storage of geographic information	GML/3.2.1
		OGC	KML is a file format used to display geographic data in an Earth browser such as Google Earth, Google Maps, and Google Maps for mobile. KML uses a tag-based structure with nested elements and attributes and is based on the XML standard	KML/2.2
		OGC	ARML (augmented reality markup language) is a specification that allows content developers create content that is displayed on various mobile AR browsers. ARML is built on a subset of KML, a set of common tags supported across all browsers and some browser specific extensions.	ARML/2.0

multiple criteria: location, direction •Built-in AR Target(POI) interaction support				
AR Data types and Structures				
• Extensible AR Target (POI) information: location, category, type, time, name, description, relationships, URLs, contact information, multimedia content		W3C	A POI can be as simple as a set of coordinates and an identifier, or more complex such as a three dimensional model of a building with names in various languages, information about open and closed hours, and a civic address. POI data has many uses including augmented reality browsers, location-based social networking games, geocaching, mapping, and navigation systems.	WD-poi- core- 20110512

Appendix D. XML and JSON Examples (Informative)

D.1 ARTargetSearch messages XML and JSON encoding examples (see 5.3.1)

D.1.1 ARTargetSearch XML example

ARTargetSearchResquest XML example:

```
<MobAR xmlns="urn:oma:xml:mobar:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ARTargetSearchRequest ARClientID="xxx" Version="1.0">
               <SearchCondition>
                             <Location>
                                           <Latitude>45.11</Latitude>
                                           <Longitude>7.67</Longitude>
                             </Location>
                             <SearchRadius>300</SearchRadius>
                             <Category>Food</Category>
               </SearchCondition>
               <FiltrationCriteria>
                             <ContentFiltration>image</ContentFiltration>
               </FiltrationCriteria>
               <ViewCondition ViewType="Ar">
                             <NumAngularSections>50</NumAngularSections>
               </ViewCondition>
    </ARTargetSearchRequest>
</MobAR>
```

ARTargetSearchResponse XML example:

```
<MobAR xmlns="urn:oma:xml:mobar:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ARTargetSearchResponse ARServerID="yyy" Version="1.0">
               <ARTargetGroup AngularSectionIndex="3" ARTargetGroupID="12">
                             <ARTargetGroupLocation>
                                 <Latitude>45.117589</Latitude>
                                 <Longitude>7.668801</Longitude>
                             </ARTargetGroupLocation>
                             <ARTarget ARTargetID="123">
                                 <ARTargetInfo>
                                     <Position>
                                           <Latitude>45.117532</Latitude>
                                           <Longitude>7.678231</Longitude>
                                     </Position>
                                     <Name>Restaurant Name</Name>
                                     <Category>Food</Category>
                                     <WebsiteUrl>www.example.com</WebsiteUrl>
                                 </ARTargetInfo>
                             </ARTarget>
                             <ARTarget ARTargetID="456">
                                 <ARTargetInfo>
                                     <Position>
                                           <Latitude>45.123478</Latitude>
                                           <Longitude>7.667842</Longitude>
                                     </Position>
                                     <Name>My Pizza</Name>
                                     <Category>Food</Category>
                                     <Phone>001122334455</Phone>
                                 </ARTargetInfo>
                             </ARTarget>
               </ARTargetGroup>
               <ARTargetGroup AngularSectionIndex="6" ARTargetGroupID="13">
                             <ARTargetGroupLocation>
                                            <Latitude>45.117589</Latitude>
                                            <Longitude>7.668801</Longitude>
                             </ARTargetGroupLocation>
                             <ARTarget ARTargetID="789">
                                 <ARTargetInfo>
                                     <Position>
                                            <Latitude>45.117532</Latitude>
                                            <Longitude>7.678231</Longitude>
                                     </Position>
                                     <Name>Secret Garden</Name>
                                     <Category>Food</Category>
                                     <WebsiteUrl>www.example2.com</WebsiteUrl>
                                 </ARTargetInfo>
                             </ARTarget>
                             <ARTarget ARTargetID="567">
                                 <ARTargetInfo>
                                     <Position>
                                            <Latitude>45.123478</Latitude>
                                            <Longitude>7.667842</Longitude>
                                     </Position>
                                     <Name>Eleven</Name>
                                     <Category>Food</Category>
                                     <Phone>0033445566</Phone>
                                 </ARTargetInfo>
                             </ARTarget>
               </ARTargetGroup>
    </ARTargetSearchResponse>
</MobAR>
```

D.1.2 ARTargetSearch JSON example

ARTargetSearchRequest JSON example:

```
{"MobAR": {
    "ARTargetSearchRequest": {
        "ARClientID": "xxx",
        "Version": "1.0",
        "SearchCondition": {
            "Location": {
                "Latitude": "45.11",
                "Longitude": "7.67"
            },
            "SearchRadius": "300",
            "Category": "Food"
        },
        "FiltrationCriteria": {
            "ContentFiltration": "image"
        },
        "ViewCondition": {
            "ViewType": "Ar",
            "NumAngularSections": "50"
        }
    }
  }
}
```

ARTargetSearchResponse JSON example:

```json
{"MobAR": {
    "ARTargetSearchResponse": {
        "ARServerID": "yyy",
        "Version": "1.0",
        "ARTargetGroup": [
            {
                "AngularSectionIndex": "3",
                "ARTargetGroupID": "12",
                "ARTargetGroupLocation": {
                    "Latitude": "45.117589",
                    "Longitude": "7.668801"
                },
                "ARTarget": [
                    {
                        "ARTargetID": "123",
                        "ARTargetInfo": {
                            "Position": {
                                "Latitude": "45.117589",
                                "Longitude": "7.668801"
                            },
                            "Name": "RestaurantName",
                            "Category": "Food",
                            "WebsiteUrl": "www.example.com"
                        }
                    },
                    {
                        "ARTargetID": "456",
                        "ARTargetInfo": {
                            "Position": {
                                "Latitude": "45.123478",
                                "Longitude": "7.667842"
                            },
                            "Name": "MyPizza",
                            "Category": "Food",
                            "Phone": "001122334455"
                        }
                    }
                ]
            },
            {
                "AngularSectionIndex": "6",
                "ARTargetGroupID": "13",
                "ARTargetGroupLocation": {
                    "Latitude": "45.117589",
                    "Longitude": "7.668801"
                },
                "ARTarget": [
                    {
                        "ARTargetID": "789",
                        "ARTargetInfo": {
                            "Position": {
                                "Latitude": "45.117532",
                                "Longitude": "7.678231"
                            },
                            "Name": "SecretGarden",
                            "Category": "Food",
                            "WebsiteUrl": "www.example2.com"
                        }
                    },
                    {
                        "ARTargetID": "567",
                        "ARTargetInfo": {
                            "Position": {
                                "Latitude": "45.123478",
                                "Longitude": "7.667842"
                            },
                            "Name": "Eleven",
                            "Category": "Food",
                            "Phone": "0033445566"
                        }
                    }
                ]
            }
        ]
    }
  }
}
```

D.2 ARContentRetrieve messages XML and JSON encoding examples (see 5.3.2)

D.2.1 ARContentRetrieve XML example

ARContentRetrieveResquest XML example:

```
<MobAR xmlns="urn:oma:xml:mobar:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ARContentRetrieveRequest ARClientID="xxx" Version="1.0" >
               <ARTargetID>789</ARTargetID>
               <FiltrationCriteria>
                             <ContentFiltration>image</ContentFiltration>
               </FiltrationCriteria>
    </ARContentRetrieveRequest>
</MobAR>
```

ARContentRetrieveResponse XML example:

```
<MobAR xmlns="urn:oma:xml:mobar:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ARContentRetrieveResponse ARServerID="yyy" Version="1.0" >
               <ARTarget ARTargetID="789">
                             <ARContent ID="2045">
                                 <Content ContentType="image/jpeg"
ContentUrl="http://example2.com/albums/pics/normal_16102008252.jpg" />
                             </ARContent>
               </ARTarget>
    </ARContentRetrieveResponse>
</MobAR>
```

D.2.2 ARContentRetrieve JSON example

ARContentRetrieveRequest JSON example:

```
{"MobAR": {
    "ARContentRetrieveRequest": {
        "ARClientID": "xxx",
        "Version": "1.0",
        "ARTarget": {
            "ARTargetID": "789",
            "FiltrationCriteria": {
                "ContentFiltration": "image"
            }
        }
    }
  }
}
```

ARContentRetrieveResponse JSON example:

```
{"MobAR": {
    "ARContentRetrieveResponse": {
        "ARServerID": "yyy",
        "Version": "1.0",
        "ARTarget": {
            "ARTargetID": "789",
            "ARContent ": {
                "ID": "2045",
                "ContentType": "image/jpeg",
                "ContentUrl": "http://example2.com/albums/pics/normal_16102008252.jpg"
            }
        }
    }
  }
}
```

D.3 ARContentSubscribe messages XML and JSON encoding examples (see 5.3.3)

D.3.1 ARContentSubscribe XML example

ARContentSubscribeResquest XML example:

```
<MobAR xmlns="urn:oma:xml:mobar:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ARContentSubscribeRequest ARClientID="xxx" Version="1.0">
               <SubscriptionRule>
                             <SearchCondition>
                                            <Location>
                                                           <Latitude>13.75</Latitude>
                                                           <Longitude>100.466667</Longitude>
                                            </Location>
                                            <SearchRadius>900</SearchRadius>
                                            <Category>Food</Category>
                             </SearchCondition>
                             <TriggerCondition>
                                            <PushInterval>PT5M30S</PushInterval>
                             </TriggerCondition>
                             <FiltrationCriteria>
                                            <ContentFiltration>image</ContentFiltration>
                             </FiltrationCriteria>
               </SubscriptionRule>
               <Duration>P7D</Duration>
    </ARContentSubscribeRequest>
</MobAR>
```

ARContentSubscribeResponse XML example:

```
<MobAR xmlns="urn:oma:xml:mobar:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ARContentSubscribeResponse ARServerID="yyy" Version="1.0">
               <SubscriptionID>sub1234</SubscriptionID>
               <SubscriptionStatus>Active</SubscriptionStatus>
               <Duration>P7D</Duration>
    </ARContentSubscribeResponse>
</MobAR>
```

In case of request to unsubscribe, the same request message is used. Unsubscribe request XML example:

```
<MobAR xmlns="urn:oma:xml:mobar:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ARContentSubscribeRequest ARClientID="xxx" Version="1.0" >
               <SubscriptionID>sub1234</SubscriptionID>
               <Duration>P0D</Duration>
    </ARContentSubscribeRequest>
</MobAR>
```

In case of request to unsubscribe, the same response message is used. Unsubscribe response XML example:

```
<MobAR xmlns="urn:oma:xml:mobar:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ARContentSubscribeResponse ARServerID="yyy" Version="1.0">
               <SubscriptionID>sub1234</SubscriptionID>
               <SubscriptionStatus>Terminated</SubscriptionStatus>
    </ARContentSubscribeResponse>
</MobAR>
```

D.3.2 ARContentSubscribe JSON example

ARContentSubscribeRequest JSON example:

```
{"MobAR": {
    "ARContentSubscribeRequest": {
        "ARClientID": "xxx",
        "Version": "1.0",
        "SubscriptionRule": {
            "SearchCondition": {
                "location": {
                    "latitude": "13.75",
                    "longitude": "100.466667"
                },
                "searchradius": "900",
                "category": "Food"
            },
            "TriggerCondition": {
                "PushInterval": "PT5M30S"
            },
            "FiltrationCriteria": {
                "ContentFiltration": "image"
            }
        },
        "Duration": "P7D"
    }
  }
}
```

ARContentSubscribeResponse JSON example:

```
{"MobAR": {
    "ARContentSubscribeResponse": {
        "ARServerID": "yyy",
        "Version": "1.0",
        "SubscriptionID": "sub1234",
        "SubscriptionStatus": "Active",
        "Duration": "P7D"
    }
  }
}
```

In case of request to unsubscribe, the same request message is used. Unsubscribe request JSON example:

```
{"MobAR": {
    "ARContentSubscribeRequest": {
        "ARClientID": "xxx",
        "Version": "1.0",
        "SubscriptionID": "sub1234",
        "Duration": "P0D"
    }
  }
}
```

In case of request to unsubscribe, the same response message is used. Unsubscribe response JSON example:

```
{"MobAR": {
    "ARContentSubscribeResponse": {
        "ARServerID": "yyy",
        "Version": "1.0",
        "SubscriptionID": "sub1234",
        "SubscriptionStatus": "Terminated"
    }
  }
}
```

D.4 ARInteractionReport message XML and JSON encoding examples (see 5.3.4)

D.4.1 ARInteractionReport XML example

ARInteractionReport XML example:

```
<MobAR xmlns="urn:oma:xml:mobar:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ARInteractionReport ARClientID="xxx" Version="1.0">
               <ARTargetID>789</ARTargetID>
               <ARInteraction>
                             <Interaction>select</Interaction>
                             <Vote>80</Vote>
                             <Timestamp>2012-09-18T21:32:52</Timestamp>
               </ARInteraction>
    </ARInteractionReport>
</MobAR>
```

D.4.2 ARInteractionReport JSON example

ARInteractionReport JSON example:

```
{"MobAR": {
    "ARInteractionReport": {
        "ARClientID": "xxx",
        "Version": "1.0",
        "ARTargetID": "789",
        "ARInteraction": {
            "Interaction": "select",
            "Vote": "80",
            "Timestamp": "2012-09-18T21:32:52"
        }
    }
  }
}
```