MobAd Technical Specification

- Core Specification

Approved Version 1.0 – 20 Mar 2012

Open Mobile Alliance

OMA-TS-MobAd_Core-V1_0-20120320-A

Contents

Figures

Tables

1. Scope

This document aims at specifying the functions and interfaces of MobAd Ad Server and MobAd Ad Engine as well as describing expected behaviors from Ad Engine, Ad Server, Ad App and SP App.

These functions and behaviors relate to:

- Ad Selection functions (e.g. based on criteria such as rules, metadata, C&P data)
- Delivery of selected Ad(s)
- Support for interactivity capabilities
- Reporting, measurement and validation of user actions with ads. (e.g. for the purpose of metrics consolidation)
- Ad pre-fetching, caching, and managing Ad
- Notification
- Applying rules and Policy

This document specifies the message flows, message patterns and data formats used for message exchanges between:

- Ad Server and Ad Engine
- Ad Server and SP App
- Ad Engine and Ad App.

This document further specifies the parameters used for Ad Metadata and Ad Metrics data by MobAd components and Ad App and SP App.

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/

[OMA-MobAd-RD] "Mobile Advertising Requirements", Open Mobile Alliance™, OMA-RD-Mobile_Advertising-V1_0, URL:http://www.openmobilealliance.org/

[OMA-MobAd-AD] "Mobile Advertising Architecture Document", Version 1.0, Open Mobile Alliance™, OMA-AD-MobAd-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

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

[DCD TSS] "Dynamic Content Delivery Technical Specification– Semantics and Transactions", Version 1.0, Open Mobile Alliance™, OMA-TS-DCD_Semantics-V1_0-20081223-C, URL:http://www.openmobilealliance.org/

[WAP-Push] "Push Over The Air", Version 2.1, Open Mobile Alliance™, OMA-WAP-TS-PushOTA-V2_1-20051122-C, URL:http://www.openmobilealliance.org/

[RFC 2445] Internet Calendaring and Scheduling Core Object Specification (iCalendar)

http://www.ietf.org/rfc/rfc2445.txt

[RFC4346] The Transport Layer Security (TLS) Protocol Version 1.1, April 2006,

URL: http://tools.ietf.org/html/rfc4346

[RFC5246] The Transport Layer Security (TLS) Protocol Version 1.2, August 2008

URL: http://tools.ietf.org/html/rfc5246

2.2 Informative References

[OMA-DICT] "OMA Dictionary", Open Mobile Alliance™, OMA-Dictionary-V2_7, URL:http://www.openmobilealliance.org/

[RFC2822] "Internet Message Format", P. Resnick, Apr 2001, URL: http://www.ietf.org/rfc/rfc2822.txt

[MMA_Guideline] "Mobile Advertising Guidelines" Oct, 2008, URL:http://mmaglobal.com/mobileadvertising.pdf

[MMA_Glossary] "Mobile Marketing Glossary", Dec, 2008, URL:http://mmaglobal.com/glossary.pdf

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.

3.1.1 Additional Conventions

Explanation of the XML Type in MobAd 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 MobAd 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

Ad App An application running on the Device which interacts with the Ad Engine in order to present Ad(s) to the user

Ad Campaign See definition for Campaign.

Ad Channel See definition in [OMA-MobAd-RD].

Ad Engine A logical component that refers to MobAd Enabler Entities on the Device as per the definition in [OMA-MobAd-RD]. The Ad Engine is an application implementing MobAd Enabler functions and running on the Device. The Ad Engine interacts with Ad App and Ad Server. See Section 5. 1 for the details

Ad Impression	See definition for Impression.
Ad Metadata	See definition in [OMA-MobAd-RD]. Examples of usage are targeted audience, capping rules, Ad expiration, etc. In the context of the OMA MobAd Core TS, Ad Metadata is a generic term and does not correspond to a specific structure or finit list of parameters.
Ad Metrics	See definition in [OMA-MobAd-RD].
Ad Selection	See definition in [OMA-MobAd-RD].
Ad Server	A logical component that refers to MobAd Enabler Entities on the Network as per the definition in [OMA-MobAd-RD].
Advertisement	See definition in [OMA-MobAd-RD].
Advertiser	See definition in [OMA-MobAd-RD].
Campaign	See definition in [OMA-MobAd-RD].
Content Metadata	See definition in [OMA-MobAd-RD] and ContextKeyword, see definition in section 5.1.
Content Provider	See definition in [OMA-MobAd-RD].
Contextualisation	See definition in [OMA-MobAd-RD]
Impression	See definition in [OMA-MobAd-RD].
MobAd Rules	The MobAd Enabler service-related metadata (as opposed to Ad metadata), that define the Service Provider's rules related to e.g. Ad prefetching and caching policies.
Personalisation	See definition in [OMA-MobAd-RD]
Principal	See definition in [OMA-MobAd-RD].
SP App	See definition in [OMA-MobAd-RD].
User	See definition in [OMA-MobAd-RD].
User Context	See definition in [OMA-MobAd-RD].

3.3 Abbreviations

Ad	Advertisement
Ad App	Ad Application. Refers to MobAd RD for full definition.
C&PR	Contextualisation and Personalisation Resources
ID	Identifier
MobAd	Mobile Advertising
OMA	Open Mobile Alliance
PAP	Push Access Protocol
SP	Service Provider
SP App	Service Provider Application. Refers to MobAd RD for full definition
URL	Universal Resource Locator
URI	Unique Resource Identifier

4. Introduction

This document defines the technical framework and specifies globally interoperable technologies for mobile advertising service over different distribution systems based on the MobAd requirement document [OMA-MOBAD-RD] and the MobAd architecture document [OMA-MOBAD-AD].

The OMA MobAd Enabler provides a set of capabilities allowing to:

- **Enable the contextualisation and personalisation of advertisements**, which is one of the key performance indicators for the effectiveness of mobile advertising.
- **Enable interactivity mechanisms with advertisements**, which improves user mobile advertising experience and enables various charging models for mobile advertising.
- **Collect and report Ad Metrics data**, which enables recording and collecting users' behaviour and impact of mobile advertising on them.

Section 5 of this document contains the details on operations and functions of components in the MobAd environment; that are Ad Server and Ad Engine (within the enabler), as well as SP App and Ad App (external actors).

Section 6 of this document contains a description of the used parameters, MobAd interface messages and common structures are defined.

Section 7 of this document contains interface message bindings over HTTP, WAP Push and possible Adaptation references.

Section 8 of this document contains details on security considerations between MobAd components i.e., Ad Engine and Ad Server (within the enabler) as well as between the MobAd components and the Ad App, SP App.

Further details for specific adaptation of MobAd over DCD and MobAd over BCAST should be referred to [MobAd-TS-DCD] and [MobAd-TS-BCAST] respectively.

4.1 Version 1.0

The MobAd Core Technical Specification addresses the requirements targeted for this release. Some features may require updates in later releases, either because requirements have been deferred to a future release (e.g. in the case of Content Scanning) or because additional use cases or detailed requirements may be needed.

Please refer to [OMA-MobAd-RD] for those features that are identified as future releases.

5. MobAd Operations and Functions

This section defines the MobAd operations and functions of the Ad Server, Ad Engine, SP App and Ad App.

5.1 Ad Engine Operations, Functions and Behaviours

5.1.1 Activating to the Ad Server

The Ad Engine can be deployed on a device via various means such as device provisioning methods, provided with the device application platform, downloaded at a later time, etc.

The Ad Engine SHALL be provided with the following information:

- Ad Server ID
- Ad Server communication addresses
- Credentials (e.g. proxy, security)

Credentials are not necessarily specific to MobAd.

The Ad Engine implementation SHOULD activate itself with the Ad Server. The activation step can be achieved, depending on SP policies, either during Ad Engine initialization, processing the first Ad Request message or by issuing a specific communication message.

At this step the Ad Server SHOULD receive the Ad Engine ID and MobAd Enabler version.

If the activation is successful, the Ad Server implementation ought to send back a response to Ad Engine to inform that the activation succeeded.

5.1.2 Pre-fetching, Caching and Managing Ads

The Ad Engine SHALL be able to support two modes of operation:

- The Ad Engine pre-fetches and caches Ads. In such case, the Ad Engine may retrieve Ads from the cache when receiving an AdRequest from the Ad App.
- The Ad Engine augments the AdRequest received from the Ad App with parameters known to the Ad Engine before sending the AdRequest to the Ad Server. In such case caching is not involved.

The Ad Engine MAY be able to receive rules related to the pre-fetching and caching of Ads. The Ad Engine implementation usually has local settings for the frequency of Ad pre-fetching, duration of Ad Caching and the size of the associated Ad storage area.

If some Ads are pre-fetched, the Ad Engine SHALL manage (i.e. filter, store, delete) Ads locally on the device according among other to the following criteria:

- Ad expiry date and time;
- Capping (e.g. Ad should not be shown more than 1 time);
- Ad Unit
- SP rules and policies applying to the particular Ad or general rules;
- Ad/Ad metadata update, if supported

The Ad Engine implementation can use local settings or data that can change over time to manage the locally cached Ads (e.g. cache size, Ad App, profile/preferences, location information).

5.1.3 Identifying and Activating Ad Apps

The Ad Engine SHOULD identify if an Ad App belongs to the Ad Server's Service Provider domain. The means to perform this identification are out of scope of MobAd and are dependant on the security mechanism of the underlying execution environment.

The identification and activation of the Ad App can be achieved in various ways, such as during the installation or initialization of the Ad App (if the given Ad App is provisioned to the device), while processing the first Ad App request, through a specific communication exchange between Ad App and Ad Engine, etc. The means to perform this function are considered out of scope of this specification.

The Ad Engine SHALL obtain the AdAppID from the Ad App.

The Ad Engine implementation can maintain a list of Ad Apps for:

- Ad Selection
- Sending Ad Request to the Ad Server.

The Ad Engine SHALL maintain Ad Apps information (and related metrics data) for metrics data reporting purposes.

5.1.4 Selecting and Returning Ads for/to an Ad App

The Ad Engine SHALL be able to select and return Ads or references to Ads from the local cache based upon the receipt of an AdRequest message from an Ad App.

The Ad Selection process uses the following information:

- Ad Request parameters;
- Ad Metadata;
- Local data;
- SP rules and policies.

Once the Ad selection process is completed, the Ad Engine SHOULD return one of the following:

- The selected Ad or reference to the Ad;
- An indication for no Ad to be returned to the Ad App.

In the case the optionally supported Ad Content Type attribute indicates Static or Dynamic Ad Content, and if the Ad Engine supports such function, the Ad Engine shall be able to combine the Dynamic and Static Ad content prior to returning the complete Ad to the Ad App.

5.1.5 Requesting and Receiving Ads from the Ad Server

The Ad Engine SHALL support requesting Ad(s) from the Ad Server.

The Ad Engine requests Ad(s) from the Ad Server in the following scenarios:

- Upon an Ad App request to the Ad Engine. The Ad Engine may determine that no Ads from the device's local storage are suitable to fulfill a corresponding Ad App request or may prefer sending the AdRequest to the Ad Server;
- The Ad cache and/or local storage is empty or below a given threshold; or,
- The variety of Ads in the cache is too limited.

If supporting the Push methods (inc. DCD, BCAST), the Ad Engine may receive Ad(s) from the Ad Server when Ad(s) becomes available at the Ad Server, according to a push schedule, according to push campaign characteristics or other SP policies or conditions. The condition types upon which Ads are pushed to the Ad Engine are undefined in MobAd.

Upon reception of Ads, the Ad Engine SHALL be able to manage (e.g. store and cache, if necessary filter) the set or a subset of the received Ads. Based on updated conditions (e.g. change in user preference, context, location, available Ad Apps etc) the operation of receiving another set of Ads, filtering and storing all or part of them can be repeated.

The Ad Engine SHALL be able to provide information as part of the AdRequest message to the Ad Server as specified in section 6.4.1.1

The Ad Engine can augment an Ad App request message that is passed to the Ad Server with additional data in order to increase the targeting parameters and help the Ad Server refine the Ad Selection process.

The Ad Engine SHALL be capable of indicating that the request for a given Ad is "urgent" by using the AdUrgency parameter. The Ad Engine implementation can determine the level of urgency of the request, e.g. based on the type of Ad App issuing the AdRequest or the level of interactivity of the Ad App with Ads, the Ad is for immediate display, etc. When using the AdUrgency parameter the Ad Engine could expect a faster answer from the Ad Server and a less targeted advertisement than if the AdUrgency parameter was not used.

The AdEngineAdRequest and the AdAppAdRequest are two different transactions and the number of Ads requested in the AdAppAdRequest is not necessarily the same as in the AdEngineAdRequest. For instance, the Ad Engine can request more Ads for other usages to the Ad Server.

5.1.6 Ad Metrics, Data Aggregation, Validation and Reporting

The Ad Engine is in charge of receiving Ad metric data information from various sources. Sources include:

- Ad Apps;
- The Ad Engine itself (e.g. metric validation information); and
- Other device components.

Ad metric data known by the Ad Engine itself could relate to:

- the management of received Ads (e.g. delay between the Ad reception and the passing to an Ad App),
- the knowledge of which Ads were passed or not passed to a given Ad App
- and Ads that were discarded from the Ad cache (e.g. expired, or as the user interest/profile changes).

The Ad Engine MAY be provided with additional data from device components such as time, location, roaming etc. This data can be used for Ad metrics validation and reports.

The Ad Engine creates a metric report by providing an aggregation of Ad metrics data from the sources described above.

The Ad Engine SHALL be able to combine multiple MetricReport structures in one message.

If the Ad Engine receives, from the Ad Server, a request for immediate metrics reporting, the Ad Engine SHALL issue a report as soon as possible.

The Ad Engine MAY by default validate Ad Metric data provided by Ad Apps, and SHALL indicate in the report if the report was validated or not and if reported information was valid or not (Default is True). The Ad Engine MAY decide not to validate Ad Metric data provided by Ad Apps e.g. when the Ad Engine determines that the Ad App is trusted (e.g. based on a certificate, based on historic accumulation of the Ad App reports, etc).

The methods used by an Ad Engine for the validation steps are out of scope of the MobAd core TS.

5.1.7 Applying Rules and Policies

The Ad Engine MAY be able to receive rules from the Ad Server and apply them to corresponding MobAd functions. If implemented, the Ad Engine does the following:

- Upon reception of a "ReportFrequency" rule, the Ad Engine SHALL use the value specified in that rule as the maximum elapsed time between consecutive metric reports. If the Ad Engine cannot send a consecutive report in the

given timeframe due to a local condition (device is off), upon resumption, the Ad Engine SHOULD immediately send a metric report.

- Upon reception of a "BatteryLevelThreshold" rule, the Ad Engine SHALL use the value specified in that rule as the threshold below which some/all MobAd functions can be suspended (e.g. postpone metric reporting if battery level is lower than 5%). The Ad Engine may set different battery thresholds to suspend different MobAd functions.
- Upon reception of an "AdCaching Threshold" rule, the Ad Engine SHOULD use the value specified in that rule as the threshold that triggers an AdEngineAdRequest, when the amount of Ads available in the local storage is below the threshold.
- Upon reception of a "SuspendResume" rule, the Ad Engine SHALL use the value specified in that rule as the triggers for which the suspension is allowed. The Ad Engine SHALL resume its functions, when returning to its previous state. The Triggering events for suspend and resume are described in section 6.4.3.2.1.
- Upon reception of a "TotalOverallCapping" rule, the Ad Engine SHALL limit the overall amount of Ads to be provided to Ad App and displayed on the device during the time defined in that rule.

5.1.8 Receiving notification from the Ad Server

The Ad Engine MAY support push notification. If supported the Ad Engine SHALL be able to receive the following notification type from the Ad Server:

- Notification for immediate cancellation of ads or campaign.
- Notification of newly available or updated rules. Upon reception of such notification, the Ad Engine SHALL issue an AdEngineRuleRequest message to the Ad Server.
- Notification for immediate reporting of Metrics. Upon reception of such notification, the Ad Engine SHALL issue an AdEngineMetricsReport message to the Ad Server as soon as possible.
- Notification of newly available Ads. Upon reception of such notification, the Ad Engine SHALL issue an AdEngineAdRequest message to the Ad Server.

5.1.9 Delivery and Network Selection

The Ad Engine SHALL support sending and receiving messages using the HTTP transport protocol.

The Ad Engine MAY support receiving messages from the Ad Server over WAP PUSH, the DCD and BCAST Enablers.

5.1.10 Suspending and Resuming MobAd functions

The Ad Engine MAY suspend and resume MobAd operations, based on local conditions that may include, battery threshold; bandwidth threshold, opt-out, overall capping, emergency call, roaming. Depending on the implementation, the Ad Engine implementation may receive notifications from, or may be a listener to, other components on the device, or may readily have such information available.

If a Service Provider provides rules, policies or event identifiers for such conditions, and if the Ad Engine supports the rule reception function, then the Ad Engine SHALL apply them on the Service Provider behalf.

5.2 Ad Server Operations, Functions and Behaviors

The purpose of the Ad Server is mainly to distribute Ads to Users. It does not do it directly, but through either SP Apps or through Ad Engines. These two entities request the Ad Server for ads, passing along contextual information to enhance its Ad Selection functions. From them, the Ad Server obtains Ad Metrics to know what they and the Users did with the provided Ads. The Ad Server has also support functions to configure Ad Engines with MobAd Rules and manage (delete/update) ads stored on them.

The Ad Server communicates with both SP App(s) residing in the Service Provider network and with Ad Engine(s) deployed on devices. This specification does not provide standardized interfaces and messages between the Ad Server and Ad App(s).

Exchanges between the Ad Server and Ad App(s) occur via the Ad Engine, or via out of band means, which are out of scope of this specification.

An Ad Server implementation may provide an interface to a Service Provider authority for the purpose of configuring the MobAd services. This is out of scope of this version of the specification.

An Ad Server implementation may communicate with other entities such as Ad-Networks and Advertisers among others for the purposes of requesting, receiving and/or storing Ads and campaigns. This is a deployment choice and is out of scope of this specification.

The principal identity SHALL be kept confidential within the SP domain. In the case the principal identity was provided to the Ad Server (e.g. by an SP App, Ad Engine, Ad App, other network resources) , it SHALL be anonymized by the Ad Server if information regarding that principal is sent to an external entity (e.g. Advertiser, Ad Network, etc).

5.2.1 Ad Server specific functions

5.2.1.1 Ad Selection, Contextualization & Personalization and Delivery

The Ad Server SHALL be able to select and deliver Ad(s) to the Ad Engine and SP App in the following scenarios:

- Based upon an Ad Request to the Ad Server
- AdServerPushAds upon support of the optional Push method toward Ad Engine

The Ad Server SHALL be able to select and deliver Ad(s) or reference(s) to Ads.

The Ad Selection process uses the following information:

- Ad Request parameters;
- Ad Metadata;
- Local data;
- C&PR
- Service Provider rules and policies.

The Ad Server SHALL be able to contextualize advertisement by adding information in the form of metadata to the ones provided by Advertisers or Ad-Networks.

In the case the AdType parameter is supported, the Ad Server can classify the Advertisement content as static, dynamic or default.

Once the Ad selection process is achieved, the Ad Server SHOULD return one of the following:

- Selected Ad(s) or reference(s) to Ad;
- An indication for no Ad to be returned.

In case the Ad Server receives an Ad Request with an urgency parameter, the Ad Server SHOULD provide a response as soon as possible.

In the case the Ad Server is able to return an Ad that matches the ContextualData of the AdRequest, in which a ContextualData Identifier was provided, the Ad Server SHALL return the Ad (or reference to it) with the ContextualDataID. In the case the Ad Server is not able to return an Ad that matches the ContextualData of the AdRequest, in which a ContextualData Identifier was provided, the Ad Server SHALL return the Ad (or reference to it) without the ContextualDataID and with other parameters qualifying the Ad (e.g. ContextKeywords, AdValidity, etc)

In the push case, the Ad Server SHALL provide ContextualData along with the Ad.

In the case the Ad Server is not able to return an Ad that matches parameters of the AdRequest and/or Contextualization and personalization data relevant for that requestor, the Ad Server MAY return a default Ad or an indication for no Ad.

Upon support of the Push method, the Ad Server may provide Ad(s) or Ad Reference(s) to the Ad Engine when Ad(s) becomes available at the Ad Server, according to a push schedule, according to push campaign characteristics or SP policies or other conditions. The condition types upon which Ads are pushed to the Ad Engine are undefined in MobAd. In the push case, the Ad Server SHALL support identifying the device (or group of devices) to which Ads needs to be pushed. The identifier can be the AdEngineID or other device identifier(s) depending on SP Policy.

5.2.1.2 Delete and Replace Ads

In some rare cases, the Ad Server SHALL be able to request an Ad or campaign provided in a previous message to be deleted and/or replaced by another Ad or campaign. This can be triggered by an Advertiser or Ad-Network request or per internal functions of the Ad Server. The trigger for the deletion and replacement of ads is out of scope of this specification.

The Ad Server can use the pull based AdResponse messages toward the Ad Engine and SP App to convey this delete and replace request.

If the optional push method or the optional notification method is supported, the Ad Server can use the AdServerPushAds message or the notification method toward the Ad Engine for the same purpose.

5.2.1.3 Metrics Data

The Ad Server SHALL be able to receive metrics data from Ad Engine(s), and SP App(s). The frequency of reporting metrics reports depends on SP Policies.

The Ad Server SHALL allow the Service Provider to define the mapping between the interactive mechanisms (ActionCode defined in 6.6.4) and the metricType identifiers defined in 6.6.1.

The SP App and the Ad App, via the Ad Engine, SHALL return to the Ad Server, in the metric report, the metric identifiers corresponding to the interactive mechanisms for which an action occurred. These identifiers can also be used by the Ad Server and/or by the Ad Engine (in the Ad App case) to validate and correlate the metric report.

The Ad Server receives metrics reports organized on a per Application basis (SP App metrics report, Ad Engine metrics reports is structured per Ad App). This may be directly used by the Ad Server to aggregate the metrics over time for each application, in order to provide that report to each application provider. The function of providing an aggregated metrics report to external entities such as application provider is out of scope of the specification.

The Ad Server SHOULD be able to aggregate the metrics data from the various sources and to create a metrics report organized on a per Ad basis. This may be used by the Ad Server to provide that report to the Ad-Network or Advertisers. This function of providing an aggregated metrics report to external entities such as Advertiser or Ad-Networks is out of scope of the specification.

Upon reception of an Ad Metric report that indicates that the metrics data were incorrect, the Ad Server SHALL perform additional metrics validation based on information contained in the metric report, data known at the Ad Server side.

Upon reception of an Ad Metric report that indicates that the metrics data were not validated, the Ad Server MAY perform additional metrics validation based on information contained in the metric report (i.e. not validated as Ad App has a trustworthiness status), information and rules known at the Ad Server.

The method used by an Ad Server for the aggregation and validation steps are out of scope of the MobAd Core TS.

Upon support of the Notification method, the Ad Server SHALL be able to request immediate metrics reporting to the Ad Engine and the Ad Engine SHALL be able to issue a report as soon as possible.

5.2.1.4 Delivery Method

The Ad Server SHALL support receiving and replying messages using the HTTP transport protocol.

The Ad Server MAY support sending messages to the Ad Engine over WAP PUSH, the DCD and BCAST Enablers.

5.2.2 Ad Server functions toward Ad Engine

5.2.2.1 Identifying and Activating Ad Engine

The Ad Server SHOULD identify if an Ad Engine belongs to the Ad Server's Service Provider domain. The identification and activation step can be achieved, depending on SP policies, either during Ad Engine initialization, processing the first Ad Request message or by issuing a specific communication message, etc. The means to perform the identification and activation are out of scope of this specification.

At this step the Ad Server SHOULD receive the following information:

- Ad Engine ID
- MobAd Enabler version

5.2.2.2 Delivering Rules and Policies

The Ad Server MAY support Rule delivery function. If implemented, the Ad Server SHALL be able to notify the Ad Engine of new available/updated rules.

The Ad Server SHALL be able to provide following rules to the Ad Engine upon receiving AdEngineRulesRequest from the Ad Engine:

- ReportFrequency
- BatteryLevel Threshold
- AdCaching Threshold
- SuspendResume
- TotalOverallCapping

5.2.2.3 Notifying the Ad Engine

The Ad Server MAY support Notification function. If implemented, the Ad Server SHALL be able to provide the following notification type to the Ad Engine:

- Notification for immediate cancellation of ads or campaign.
- Notification of new available/updated rules.
- Notification for immediate reporting of Metrics. Upon reception of such notification, the Ad Engine SHALL issue an AdEngineMetricReport message to the Ad Server.
- Notification of new available Ads.

5.2.3 Ad Server functions toward SP App(s)

5.2.3.1 Selecting and Delivering Ad(s) to the SP App

The Ad Server SHALL be able to select and deliver Ad(s) or reference(s) to Ads based upon the receipt of an SPAppAdRequest message from a SP App.

5.3 Ad App Operations, Functions and Behaviours

The Ad App is an external actor communicating with the Ad Engine of MobAd Enabler.

The sections below describe the expected behaviors of the Ad App when interacting with the Ad Engine.

5.3.1 Identifying Ad Apps

The Ad App is a registered application within the SP domain. This enables the Ad Engine to accept the Ad Request; select and deliver appropriate targeted Ads to those Ad Apps.

However, this is considered out of scope of this specification.

5.3.2 Requesting and Receiving Ads from the Ad Engine

The Ad App requests Ad(s) from the Ad Engine that is residing on the same device.

The Ad App can request Ad(s) from the Ad Engine based on triggering of internal execution logic within the Ad App. The Ad Request contains information as defined in Section 6.2.1.1. The Ad App can pass relevant keywords describing its contexts (e.g. content metadata, application specific information) in its request to the ad engine. The keywords (list or rules on how to define them) are defined by and subject to SP policy.

After the Ad selection process executed by Ad Engine, the Ad App receives a selected Ad or indication of no Ad.

The Ad Response message contains information as defined in Section 6.2.1.2.

The Ad App is responsible for presenting to the user the selected Ad.

5.3.3 Interaction mechanism, Collecting Impression, Reporting Metrics to Ad Engine

The support of interactivity mechanisms identified in section 6.6.4 by devices depends primarily upon a specific Service Provider policy, as well as on the level of integration between the Ad App and the underlying device platform. The supported list of actions and associated codes are provided by the SP provider to the developers of Ad Apps by out of band means.

After the Ads of interest are presented, the Ad impressions and user interactions with an Ad (e.g. click-through) is captured and reported by the Ad App to Ad Engine. . If an Ad is provided with interaction mechanisms that are not supported, by the Ad App or available device APIs, the related events will be ignored by the Ad App.

Such Ad Metric data is reported as defined in Section 6.2.2.1.

It is assumed that the Ad App immediately reports to the Ad Engine and does not aggregate the metrics for a later reporting.

5.4 SP App Operations, Functions and Behaviours

The SP App is an external actor communicating with the Ad Server of the MobAd Enabler.

The sections below describe the expected behaviours of the SP App when interacting with the Ad Server.

5.4.1 Deployment toward the Ad Server.

SP App is deployed within the service provider domain. The attribution of the SPAppID as well as the policies related to the security mechanisms to be set in place between it and the Ad Server are considered implementation and deployment specific and are out of scope of this specification.

The SP App needs to be provided with the AdServerID and the Ad Server communication addresses.

5.4.2 Identifying the principal

The SP App can be capable of identifying the principal (e.g. MSISDN, username) for which an Ad is to be delivered.

In the scope of this specification a principal is not an Ad Engine and the SP App does not communicate with Ad Engine.

The SP App can send ads to a MobAd device, but does not communicate with the resident Ad Engine using MobAd's protocols, as it is out of scope of this specification.

5.4.3 Requesting Ads from the Ad Server

Communication between SP App and Ad Server are based on a single request followed by a single response in a single session.

An SP App can be capable of requesting multiple Ads in a single request. It SHALL provide the Ad Server with its SPAppID.

As part of the AdRequest, the SP App can include Contextual Data that relates to its content (with which the Ad will be inserted) and/or to the user context. The various means for the SP App to retrieve such information are out of scope of this specification.

The SP App can be capable of indicating that the request for a given Ad is “urgent” by using the AdUrgency parameter. The SP App implementation can determine the level of urgency of the request, e.g. based on the level of interactivity of the Content to which the Ad is to be associated, the Ad is for immediate display, etc. When using the AdUrgency parameter the SP App could expect a faster answer from the Ad Server and a less targeted advertisement than if the AdUrgency parameter was not used.

5.4.4 Managing Ads

Upon reception of Ads or reference to Ads from the Ad Server, the SP App should apply the rules and parameters provided by the Ad Server, in the SPAppAdResponse (e.g. capping, preservedAdFormat, etc).

An SP App implementation may store the received Ads and related parameters. In case an implementation support storing Ads, it should manage (e.g., cache and delete) Ads with regards to the following:

- Ad expiry date and time;
- Capping (e.g. Ad should not be shown more than 1 time per requestor/principal);
- Ad format;
- SP rules and policies applying to the particular Ad, to a particular type of Ad or general rules;
- Local settings or data that can change over time (e.g. discarded content)
- Ad deletion and replacement,
- etc.

5.4.5 Metrics reporting

The SP App is capable of aggregating and reporting metrics to the Ad Server in a single transaction.

5.4.6 Applying rules and policies

SP App is governed by Service Provider rules and policies.

6. Interfaces (Normative)

6.1 Description of Frequently Used Parameters

The following table contains descriptions of frequently used parameters used throughout the document:

Parameter name	Data type	Description
AdAppID	String	SP-unique identifier of the Ad App (i.e. Identifies the application, as opposed to the application instance) , e.g. “soccx21330001”. Note: the assignment and generation of AdAppID are out of scope **Used in**: AdAppAdRequest; AdAppMetricsReport; AdEngineAdRequest; AdEngineAdResponse; AdEngineMetricsReport; AdServerPushAds
AdUnit	String	Provides information regarding the Ad media type such as MIME type, format, size for the Ads, e.g. “MMA_v1.2_AdUnit_017” or Image/JPEG_240x120”. Note: Mobile Advertising Guidelines defined by MMA dated as Oct, 2008 may be used to determine the IOP test cases. The term Ad Unit corresponds to MMA Glossary term [MMA Glossary]. Ad Currency could be used for additional purposes at the discretion of the Service Provider and subject to SP Policy, e.g. , “CMCC_V1.0_AdUnit_011”. **Used in**: AdAppAdRequest; SPAppAdRequest; AdEngineAdRequest; AdEngineAdResponse; AdServerPushAds
AdEngineID	String	SP-unique identifier of the Ad Engine and hence implicitly target user. For example, “MSISDN +447919994675” or “AdE#1234567x89”. Note: the assignment and generation of the AdEngineID is out of scope. **Used in**: AdEngineAdRequest; AdEngineMetricsReport; AdEngineRulesRequest
AdID	String	SP-unique identifier of the concerned instance of the Ad, e.g. “acme_co_2133x0001”. It contains the campaign ID. The AdID shall be composed of a first sequence of characters identifying a campaign whereas the last two (2) characters of the string denote the ad storyboarding order within the campaign, as hexadecimal values. **Used in**: AdAppAdResponse; SPAppAdResponse; AdEngineAdResponse; AdServerPushAds
AdServerID	String	Identifies the Ad Server. **Used in**: AdEngineRulesResponse; AdServerPushAds; AdServerPushRules; AdServerPushNotif
AdValidity	String	The expected period of the lifetime of the Ad, formatted as per RFC 2445 **Used in**: SPAppAdRequest; SPAppAdResponse; AdEngineAdRequest; AdEngineAdResponse; AdServerPushAds
ContextualDataID	String	Identifies the ContextualData structure that was passed in the SP App or Ad App request and that is relevant to this Ad. **Used in**: AdAppAdResponse; SPAppAdResponse; ContextualData

MessageID	String	A unique identifier that is given by the application (i.e. Ad App, Ad Engine or SP App) in the request. The response to a given request MUST contain the same MessageID. Note: how to generate the MessageID is out of scope. **Used in**: AdEngineAdRequest; AdEngineAdResponse; AdEngineMetricsReport; AdEngineRulesRequest; AdEngineRulesResponse
RequestUrgency	Boolean	Indicates the urgency of the request. If the parameter is not sent: 1. if the source of the request is an Ad Engine, the Ad Server SHALL interpret the absence of this parameter as FALSE (no urgency). 2. if the source of the request is an SP App, the Ad Server SHALL interpret the absence of this parameter as TRUE (urgency). If the sender wants to specify an urgency, a value of RequestUrgency=FALSE can be sent to indicate a lack of urgency for the response (e.g. for the case of pre-fetching Ads), while a value of RequestUrgency=TRUE can be sent to indicate the need for an immediate response. **Used in**: SPAppAdRequest; AdEngineAdRequest
SPAppID	String	SP-unique identifier of the concerned SP App, e.g. “portal01”. Note: how to generate the SPAppID is out of scope. **Used in**: SPAppAdRequest; SPAppMetricsReport
Version	String	Indicates the MobAd version. **Used in**: SPAppAdRequest; AdEngineAdRequest; AdEngineRulesRequest

Table 1: Frequently used parameters

Parameter name	Data type	Description
AdContentData	Structure	Contains the actual Ad content payload which includes associated formatting parameters (e.g. MIME type, encoding). **Used in**: AdAppAdResponse; SPAppAdResponse; AdEngineAdResponse; AdServerPushAds
AdUsage	Structure	Contains the parameters related to the usage of each Ad (e.g. capping, expiry date) **Used in**: SPAppAdResponse; AdEngineAdResponse; AdServerPushAds
AdMetricsData	List of Structures	List of structures of which each structure defined user metrics of a single type. **Used in**: AdAppMetricsReport; SPAppMetricsReport; AdEngineMetricsReport
ContextualData	Structure	Contains data describing the Ad opportunity context and facilitating Ad selection such as Content Metadata, location, etc (e.g. “car game”). Contextual data should represent the context relevant to the target application and/or content. **Used in**: AdAppAdRequest; SPAppAdRequest; AdEngineAdRequest;

		AdEngineAdResponse; AdServerPushAds

Table 2: Frequently used structures

6.2 MobAd-1 interface

MobAd-1 is an interface exposed by the Ad Engine to the Ad App. The Ad App uses this interface to request and obtain Ads and their associated Ads identifiers from the Ad Engine, as well as to report Ad Metrics data to the Ad Engine, accompanied by the associated Ads identifiers.

The MobAd-1 interface allows the Ad Engine to:

- Receive the Ad Request from Ad App
- Provide Ad Response to the Ad App
- Receive the reported Ad Metrics data from the Ad App

The MobAd-1 interface allows the Ad App to:

- Request and obtain Ads from the Ad Engine
- Report Ad Metrics data to the Ad Engine

There is one request, followed by one response, each of which contains a number of required parameters, as described below.

6.2.1 Ad App - Ad Request and Ad Response

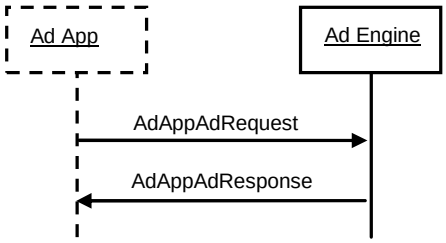

Figure 1: Flow of Ad App Ad Request, and its response.

Message	Requirement	Direction
AdAppAdRequest	Mandatory	Ad App → Ad Engine
AdAppAdResponse	Mandatory	Ad App ← Ad Response

6.2.1.1 AdAppAdRequest message

This message is used by the Ad App to request a single Ad from Ad Engine.

It is assumed that the device runtime environment is responsible for correlating of Ad Requests and Ad Responses.

In the Ad Request, the Ad App SHALL be able to pass the parameters as described in the first column of Table 2, and the said parameters SHOULD be of the type denoted in the second and third column of Table 2. The cardinality of each parameter in a given request SHALL be as denoted in the fourth column of Table 2.

In contrast, the fifth column in Table 2 is given solely for informative purposes.

Parameter name	XML Type	Data type	Parameter cardinality	Description
AdAppAdRequest	E		1	Indicate the name of the message

AdAppID	A	String	1	See description in section 6.1.
AdUnit	A	String	0..1	See description in section 6.1.
ContextualData	E1	Structure	0..N	See description in section 6.1.

Table 3: AdAppAdRequest message parameters

6.2.1.2 AdAppAdResponse message

This message is used by the Ad App to receive a single Ad to from Ad Engine.

In the Ad Response, the Ad Engine SHALL be able to pass the parameters as described in the first column of Table 3, and the said parameters SHOULD be of the type denoted in the second and third column of Table 3. The cardinality of each parameter in a given request SHALL be as denoted in the fourth column of Table 3.

In contrast, the fifth column in Table 3 is given solely for informative purposes.

Parameter name	XML Type	Data type	Parameter cardinality	Description
AdAppAdResponse	E		1	Indicates the name of the message
AdID	A	String	1	See description in section 6.1.
ContextualDataID	A	Comma separated list of strings.	0..1, Conditional	See description in section 6.1. Condition: This parameter is returned if Ad matches one of multiple contexts specified in the request. It SHALL be present if more than one context was provided in the request.
AdContentData	E1	Structure	1	See description in section 6.1.
PreserveAdFormat	A	Boolean	0..1	When TRUE Ad format MUST be preserved. Note: Ad Engine SHOULD pass this through to the Ad App as an indication whether format change is allowed (e.g. mobile gaming).

Table 4: AdAppAdResponse message parameters

Note: When the AdAppAdResponse contains only the parent element, it SHALL be considered as a No Ad.

6.2.2 Ad App - Ad Metrics Data Report

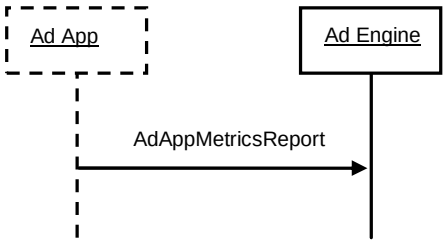

Figure 2: Flow of Ad App Ad Metrics data report

Message	Requirement	Direction
AdAppMetricsReport	Mandatory	Ad App → Ad Engine

6.2.2.1 AdAppMetricsReport message

This message is used by the Ad App to report Ad Metrics data to the Ad Engine.

It is assumed that the Ad Engine implementation is responsible for acknowledgement of Ad Metrics data report.

In the AdAppMetricsReport message, the Ad App SHALL be able to pass the parameters as described in the first column of Table 4, and the said parameters SHOULD be of the type denoted in the second and third column of Table 4. The cardinality of each parameter in a given message SHALL be as denoted in the fourth column of Table 4.

Parameter name	XML Type	Data type	Parameter cardinality	Description
AdAppMetricsReport	E		1	Indicate the name of the message.
AdAppID	A	String	1	See description in section 6.1. Note: Indicates the Ad App reporting the metrics.
AdMetricsData	E1	Structure	1..N	See description in section 6.1. Each AdMetricsData corresponds to one Ad. There can be multiple Ad metrics reported at the same time.

Table 5: AdAppMetricsReport message parameters

6.3 MobAd-2 interface

MobAd-2 is an interface exposed by the Ad Server to the SP App. The SP App uses this interface to request and obtain Ad(s), reference(s) to Ad(s), associated Ad(s) identifiers and possibly additional information, as well as to report Ad Metrics data, accompanied by the associated Ad(s) ID(s).

This interface can also be used by the Ad Server to inform the SP App that some Ad(s) (stored locally by the SP App) are supposed to be deleted. This can be achieved either by attaching Ad deletion information to an Ad Server response following an SP App request for Ad(s).

The message pattern supported by MobAd-2 is a request issued by SP App towards Ad Server followed by a synchronous response from Ad Server to SP App.

The MobAd-2 interface allows the Ad Server to:

- Receive the Ad Request from SP App
- Provide Ad Response to the SP App
- Receive the report Ad Metrics data from the SP App
- Notifies SP App to delete some Ads stored with it

The MobAd-2 interface allows the SP App to:

- Request and obtain Ads from the Ad Server
- Report Ad Metrics data to the Ad Server

6.3.1 SP App - Ad Request and Ad Response

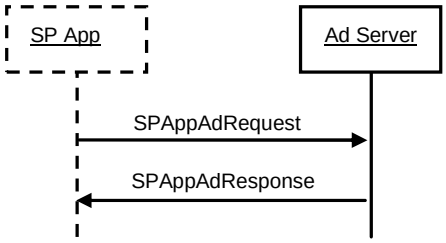

Figure 3: Flow of SP App Ad Request, and its Ad Response.

Message and Information Elements

Message	Requirement	Direction
SPAppAdRequest	Mandatory	SP App → Ad Server
SPAppAdResponse	Mandatory	SP App ← Ad Server

6.3.1.1 SPAppAdRequest message

This message is used by the SP App to request Ad(s) from Ad Server.

In the Ad Request, the SP app SHALL be able to pass the parameters as described in the first column of Table 5, and the said parameters SHALL be of the type denoted in the second and third column of Table 5. The cardinality of each parameter in a given request SHALL be as denoted in the fourth column of Table 5.

In contrast, the fifth column in Table 5 is given solely for informative purposes.

Parameter name	XML Type	Data type	Parameter cardinality	Description
SPAppAdRequest	E		1	Indicates the name of the message.
Version	A	String	0..1	See description in section 6.1.
SPAppID	A	String	1	See description in section 6.1.
RequestUrgency	A	Boolean	0..1	See description in section 6.1.
PrincipalID	A	String	0..1	Contains an identifier of the target principal, i.e. a user or a group. For example, “MSISDN +447919994675”, “userGroup-01234”, “OMA-CD@OMA.ORG”, etc. Note: PrincipalID is used by MobAd SP App, but creation of PrincipalID is out of scope.
SPAppAdRequestInfo	E1	Structure	1..N	Characterise each requested Ad.
AdUnit	A	String	0..1	See description in section 6.1.
ProvideAdContent	A	Boolean	0..1	When **True** the Ad content is returned within the response.

				When **False** an Ad URI is returned within the response. If omitted the Ad Server will decide what to provide.
AdValidity	A	String	0..1	See description in section 6.1.
ContextualData	E2	Structure	0..N	See description in section 6.1.

Table 6: SPAppAdRequest message parameters

6.3.1.2 SPAppAdResponse message

This message is used by the SP App to receive Ad(s) to from Ad Server.

In the Ad Response, the Ad Server SHALL be able to pass the parameters as described in the first column of Table 6, and the said parameters SHALL be of the type denoted in the second and third column of Table 6. The cardinality of each parameter in a given request SHALL be as denoted in the fourth column of Table 6.

In contrast, the fifth column in Table 6 is given solely for informative purposes.

Parameter name	XML Type	Data type	Parameter cardinality	Description
SPAppAdResponse	E			Indicate the name of the message.
SPAppProvidedAd	E1	Structure	0..N	Characterise each provided Ad.
AdID	A	String	0..1	See description in section 6.1.
Outdated-AdID	A	String	0..1	AdID of an outdated Ad that needs to be discarded and replaced with provided Ad, if any: 1. if present and if other parameters in the message are present in the SPAppAdResponse message, SHALL be used to replace the outdated content (Ad or parameters) 2. if present and other parameters are not, SHALL be used to delete the corresponding content (Ad and associated parameters) Condition: Present if an existing Ad needs to be updated or deleted.
ContextualDataID	A	Comma separated list of strings	0..1	See description in section 6.1. Condition: This parameter is returned if Ad matches one of multiple contexts specified in the request. It SHALL be present if more

				than one context was provided in the request.
AdUnit	A	String	0..1	See description in 6.1
AdContentData	E2	Structure	1	See description in section 6.1. Condition: 1. The Ad Server returned at least one Ad.
AdUsage	E2	Structure	0..1	See description in section 6.1

Table 7: SPAppAdResponse message parameters

In the Ad Response, the Ad Server MAY be able to pass the parameters as described in the first column of Table 7, and the said parameters SHALL be of the type denoted in the second and third column of Table 7. The cardinality of each parameter in a given request SHALL be as denoted in the fourth column of Table 7.

In contrast, the fifth column in Table 7 is given solely for informative purposes.

Parameter name	XML Type	Data type	Parameter cardinality	Description
AdType	A	Integer	0..1	Indicate if the Ad Content is of type: 0-Default 1-Static 2-Dynamic 3-256 reserved for future use.

Table 8: SPAppAdResponse message optional parameters

6.3.2 SP App - Ad Metrics Data Report

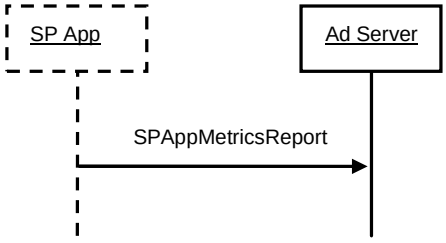

Figure 4: Flow of SP App Ad Metrics data report

Message	Requirement	Direction
SPAppMetricsReport	Mandatory	SP App → Ad Server

6.3.2.1 SPAppMetricsReport message

This message is used by the SP App to report Ad Metrics data to the Ad Server.

In the SPAppMetricsReport message, the SP App SHALL be able to pass the parameters as described in the first column of Table 8, and the said parameters SHALL be of the type denoted in the second and third column of Table 8. The cardinality of each parameter in a given message SHALL be as denoted in the fourth column of Table 8.

Parameter name	XML Type	Data type	Parameter cardinality	Description
SPAppMetricsReport	E			Indicate the name of the message.
SPAppID	A	String	1	See description in section 6.1.
AdMetricsData	E1	Structure	1..N	See description in section 6.1.

Table 9: SPAppMetricsReport message parameters

6.4 MobAd-3 interface

MobAd-3 is an interface exposed by the Ad Server to the Ad Engine. The Ad Engine uses this interface to request and obtain Ad(s), reference(s) to Ad(s), their associated Ad(s) ID(s) and Ad Metadata from the Ad Server, as well as to report Ad Metrics data to the Ad Server, accompanied by the associated Ad(s) ID(s).

This interface can also be used by the Ad Server to inform the Ad Engine that some Ad(s) (stored locally by the Ad Engine) are supposed to be deleted. This can be achieved either by attaching Ad deletion information to an Ad Server response following an Ad Engine request for Ad(s), or by returning such information in response to a specific request for updates on Ad's validity.

This interface may also be used by the Ad Engine to retrieve MobAd Rules and to provide notification to the Ad Server.

The MobAd-3 interface allows the Ad Server to:

- Receive the Ad Request from Ad Engine
- Provide Ad Response to the Ad Engine
- Receive the Ad Metrics data report from the Ad Engine
- Notify the Ad Engine to delete some Ads stored with it
- Provide MobAd Rules to the Ad Engine

The MobAd-3 interface allows the Ad Engine to:

- Request and obtain Ads from the Ad Server
- Report Ad Metrics data to the Ad Server
- Provide notification to the Ad Server
- Retrieve MobAd Rules from the Ad Server

6.4.1 Ad Engine - Ad Request and Ad Response

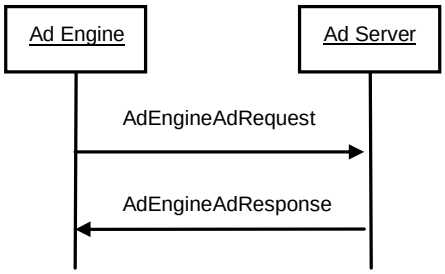

Figure 5: Flow of Ad Engine Ad Request, and its response.

Messages:

Message	Requirement	Direction
AdEngineAdRequest	Mandatory	Ad Engine → Ad Server
AdEngineAdResponse	Mandatory	Ad Engine ← Ad Server

6.4.1.1 AdEngineAdRequest message

This message is used by the Ad Engine to request Ad(s) from Ad Server.

In the Ad Request, the Ad Engine SHALL be able to pass the parameters as described in the first column of Table 9, and the said parameters SHALL be of the type denoted in the second and third column of Table 9. The cardinality of each parameter in a given request SHALL be as denoted in the fourth column of Table 9.

In contrast, the fifth column in Table 9 is given solely for informative purposes.

Parameter name	XML Type	Data type	Parameter cardinality	Description
AdEngineAdRequest	E		1	Indicate the name of the message.
Version	A	String	0..1	See description in section 6.1.
AdEngineID	A	String	1	See description in section 6.1.
MessageID	A	String	1	See description in section 6.1.
RequestUrgency	A	Boolean	0..1	See description in section 6.1.
AdEngineAdRequestInfo	E1	Structure	1..N	Characterise each requested Ad.
AdAppID	A	String	0..1	See description in section 6.1.
AdValidity	A	String	0..1	See description in section 6.1.
AdUnit	A	Comma separated list of strings	0..1	See description in section 6.1. Multiple AdUnits can be provided for each requested Ad. Each value will be indicated in the comma separated list of strings
ContextualData	E2	Structure	1..N	See description in section 6.1.

Table 10: AdEngineAdRequest message parameters

6.4.1.2 AdEngineAdResponse message

This message is used by the Ad Engine to receive Ad(s) to from Ad Server.

In the Ad Response, the Ad Server SHALL be able to pass the parameters as described in the first column of Table 10, and the said parameters SHALL be of the type denoted in the second and third column of Table 10. The cardinality of each parameter in a given request SHALL be as denoted in the fourth column of Table 10.

In contrast, the fifth column in Table 10 is given solely for informative purposes.

Parameter name	XML Type	Data type	Parameter cardinality	Description
AdEngineAdResponse	E		1	Indicate the name of the message.

MessageID	A	String	1	See description in section 6.1.
AdEngineProvidedAd	E1	Structure	0..N	Characterise each provided Ad
AdID	A	String	0..1 Conditional	See description in section 6.1. Condition: AdID is not present when the message is designed to delete an Ad without replacement.
Outdated-AdID	A	String	0..1	AdID of an outdated Ad that needs to be discarded and replaced with provided Ad, if any: 1. if present and if other parameters in the message are present in the AdEngineAdResponse message, SHALL be used to replace the outdated content (Ad or parameters) 2. if present and other parameters are not, SHALL be used to delete the corresponding content (Ad and associated parameters) Condition: Present if an existing Ad needs to be updated or deleted.
AdUnit	A	Comma separated list of strings	0..1	See definition in section 6.1 Multiple AdUnits can be provided for each Ad. Each value will be indicated in the comma separated list of strings
AdAppID	A	String	0..1	See description in section 6.1. Note: if it was present in the request, the same AdAppID SHALL be present in the response.
AdContentData	E2	Structure	0..1 conditional	See description in section 6.1. Condition: AdContentData is not present when the message is designed only to delete an Ad without replacement.
ContextualData	E2	Structure	0..N conditional	Further to the description in section 6.1: If ContextualData was provided in the corresponding AdEngineAdRequest message and successfully matched, the contextual data structure SHALL only contain the ContextualDataID parameter and no ContextKeyword parameters. If ContextualData was not provided or could not be matched, then the structure SHALL NOT contain the ContextualDataID parameter and MAY contain the ContextKeyword parameter. Condition: ContextualData is not present when the message is designed only to delete an Ad without replacement. See structure for ContextualData in section 6.6.2.

AdUsage	E2	Structure	0..1	See description in section 6.1 AdUsage is an optional parameter and SHALL not be present when the message is designed only to delete an Ad without replacement.

Table 11: AdEngineAdResponse message parameters

Note: An indication of No Ads takes the form of an AdEngineAdResponse message containing only the parent elements.

In the Ad Response, the Ad Server MAY be able to pass the parameters as described in the first column of Table 11, and the said parameters SHALL be of the type denoted in the second and third column of Table 11. The cardinality of each parameter in a given request SHALL be as denoted in the fourth column of Table 11.

In contrast, the fifth column in Table 11 is given solely for informative purposes.

Parameter name	XML Type	Data type	Parameter cardinality	Description
AdType	A	Integer	0..1	Indicate if the Ad Content is of type: 0-Default 1-Static 2-Dynamic 3-256 reserved for future use.

Table 12: AdEngineAdResponse message optional parameters

6.4.2 Ad Engine - Ad Metrics Data Report

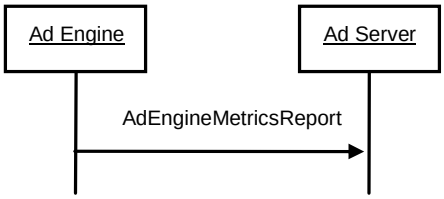

Figure 6: Flow of Ad Engine Ad Metrics data report.

Message and Information Elements:

Message	Requirement	Direction
AdEngineMetricsReport	Mandatory	Ad Engine → Ad Server

6.4.2.1 AdEngineMetricsReport message

This message is used by the Ad Engine to report Ad Metrics data to the Ad Server.

It is assumed that the HTTP 1.1 which is used as the underlying protocol is responsible for acknowledgement of Ad Metrics data report.

In the AdEngineMetricsReport message, the Ad Engine SHALL be able to pass the parameters as described in the first column of Table 12, and the said parameters SHALL be of the type denoted in the second and third column of Table 12. The cardinality of each parameter in a given message SHALL be as denoted in the fourth column of Table 12.

Parameter name	XML Type	Data type	Parameter cardinality	Description
AdEngineMetricReport	E		1	Indicate the name of the message
MessageID	A	String	1	See description in section 6.1.
AdEngineID	A	String	1	See description in section 6.1.
AdEngineMetricsData	E1	Structure	1..N	Indicates the Ad Metrics data information reported by the Ad Engine.
AdAppID	A	String	1	Refer to description in Section 6.1
ValidatedByAdEngine	A	Boolean	1	Indicates whether Ad Metrics Data information reported by Ad App was validated by the Ad Engine. Default Value: TRUE ("validated")
ValidationResult	A	Boolean	1	Indicate if the results of the Ad App metric data validation by Ad Engine are correct or incorrect. Default Value: TRUE ("passed validation") Note: Though this parameter is logically conditional, it doesn't make sense to define conditional Boolean attributes in XML.
AdMetricsData	E2	Structure	1..N	See description in section 6.1. Composition of this structure is defined in section 5.6.1.

Table 13: AdEngineMetricsReport message parameters

6.4.3 Ad Engine - MobAd Rules Request and Response

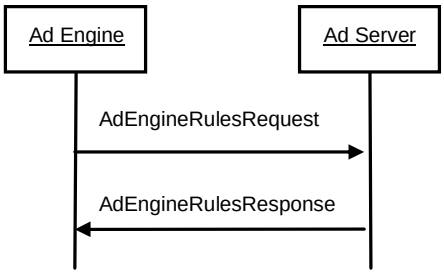

Figure 7: Flow of Ad Engine MobAd Rules request, and its response

Message and Information Elements:

Message	Requirement	Direction
AdEngineRulesRequest	Optional	Ad Engine → Ad Server
AdEngineRulesResponse	Optional	Ad Engine ← Ad Server

6.4.3.1 AdEngineRulesRequest message

This message is used by the Ad Engine to request MobAd Rules from Ad Server.

In the AdEngineRulesRequest message, the Ad Engine SHALL be able to pass the parameters as described in the first column of Table 13, and the said parameters SHALL be of the type denoted in the second and third column of Table 13. The cardinality of each parameter in a given message SHALL be as denoted in the fourth column of Table 13.

Message format is as follows:

Parameter name	XML Type	Data type	Parameter cardinality	Description
AdEngineRulesRequest	E		1	Indicate the name of the message
MessageID	A	String	1	See description in section 6.1.
Version	**A**	String	0..1	See description in section 6.1.
AdEngineID	A	String	1	See description in section 6.1.

Table 14: AdEngineRulesRequest message parameters

6.4.3.2 AdEngineRulesResponse message

This message is used by the Ad Engine to receive rule(s) from Ad Server.

In the AdEngineRulesResponse message, the Ad Server SHALL be able to pass the parameters as described in the first column of Table 14, and the said parameters SHALL be of the type denoted in the second and third column of Table 14. The cardinality of each parameter in a given message SHALL be as denoted in the fourth column of Table 14.

Message format is as follows:

Parameter name	XML Type	Data type	Parameter cardinality	Description
AdEngineRulesResponse	E		1	Indicate the name of the message
MessageID	A	String	1	See description in section 6.1.
AdServerID	A	String	1	See description in section 6.1.
MobAdRulesInfo	E1	Structure	0..N	Indicates the MobAd Rules information, for each rule, made available to the Ad Engine.

Table 15: AdEngineRulesResponse message parameters

6.4.3.2.1 MobAdRulesInfo

In the MobAdRulesInfo sent, the Ad Server SHALL be able to pass the parameters as described in the first column of Table 15, and the said parameters SHALL be of the type denoted in the second and third column of Table 15. The cardinality of each parameter in a given message SHALL be as denoted in the fourth column of Table 15.

Parameter name	XML Type	Data type	Parameter	Description

			cardinality	
RuleID	A	String	1	Unique identifier for the MobAd Rule defined by Ad Server. If the Ad Server sends the rule with an existing RuleID, that rule SHALL replace the existing rule.
RuleType	A	Integer	1	1: ReportFrequency 2: BatteryThreshold 3: AdCachingThreshold 4: SuspendResume 5:TotalOverallCapping 6-127: TBD for further usage 128 – 255: for private usage
RuleValue	A	String	1	If the rule type is "1: ReportFrequency", this value means the maximum amount of time (in minutes) between two Ad Engine metrics reports to the Ad Server. If the value is set to "0", the Ad Engine should report the metrics data without delay or interval. If the rule type is "2: BatteryLevelThreshold", this value represents the lower limit (in %) for the device battery level where device is able to perform the Ad Engine's functions (e.g. reporting to the Ad Server. (e.g. 10%).) If the rule type is "3: AdCachingThreshold", this value represents the lower limit (in KB) of amount of ads cached in the device. Amount of Ads cached in the device should not fall short of the threshold drawn. If the Rule Type is "4: SuspendResume", this value represents the list (comma-separated tokens) identifying device/user related events (e.g. roaming, emergency/911, etc.) allowed to trigger suspension of advertising. When the previous state is reached again, advertising functions are resumed. If the Rule Type is "5:TotalOverallCapping", this value represent the number of Ads per day for a user. (Example: 25 Ads per hour)

Table 16: MobAdRulesInfo parameters

6.5 Delv-1 Interface

Delv-1 is an optionally supported interface exposed by the Ad Engine. If implemented all the messages described in the below sections SHALL be supported. The Ad Engine receives Ad(s) and/or Ad Metadata over this interface from the Ad Server via underlying push and/or broadcast delivery mechanisms. The Ad Server uses this interface to push either Ad(s) or notification that Ad(s) are available for retrieval.

The Ad Server may also use this interface to provide service notification to the Ad Engine (e.g. information that SP caching and pre-fetching policies have been dynamically updated; An Ad or campaign needs to be cancelled immediately, MobAd Rules have changed, request for Ad Metrics data reporting, etc).

The Delv-1 interface allows the Ad Engine to:

- Receive Ads and/or Ad Metadata from Ad Server
- Receive MobAd Rules from Ad Server

The Delv-1 interface allows the Ad Server to:

- Push (unicast) either Ads or notifications of Ads for retrieval to Ad Engine
- Push (unicast) MobAd Rules to the Ad Engine
- Broadcast Ads and/or Ad Metadata to the Ad Engine
- Broadcast usage-related MobAd Rules to the Ad Engine
- Provide service notification to Ad Engine

6.5.1 Server-side – Push Ads

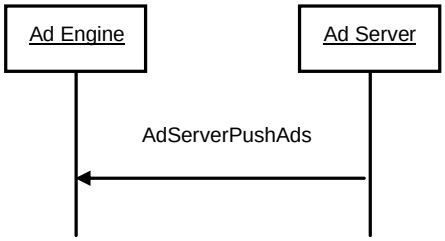

Figure 8: Flow of Ad Server Push Ads to Ad Engine

Message and Information Elements

Message	Requirement	Direction
AdServerPushAds	Mandatory	Ad Server → Ad Engine

6.5.1.1 AdServerPushAds message

This message is used by the Ad Server to push Ad(s) to Ad Engine.

In the AdSrvPushAds, the Ad Server SHALL be able to pass the parameters as described in the first column of Table 16, and the said parameters SHALL be of the type denoted in the second and third column of Table 16. The cardinality of each parameter in a given message SHALL be as denoted in the fourth column of Table 16.

Message format is as follows:

Parameter name	XML Type	Data type	Parameter cardinality	Description
AdServerPushAds	E			Indicate the name of the message.
AdServerID	A	String	1	See description in section 6.1.
PushProvidedAd	E1	Structure	1..N	Characterise each provided Ad
AdID	A	String	1	See description in Section 6.1
Outdated-AdID	A	String	0..1	AdID of an outdated Ad that needs to be discarded and replaced with provided Ad, if any:

				1- if present and if other parameters in the message are present in the AdServerPushAd message, it SHALL be used to replace the outdated content (Ad or parameters) 2- if present and other parameters are not, it SHALL be used to delete the corresponding content (Ad and associated parameters) Condition: Present if an existing Ad needs to be updated or deleted.
AdAppID	A	String	0..1	See description in section 6.1.
AdUnit	A	Comma separated list of strings	0..1	See description in table 6.1. Multiple AdUnits can be provided for each Ad. Each value will be indicated in the comma separated list of strings.
ContextualData	E2	Structure	1..N	See description in section 6.1.
AdContentData	E2	Structure	1	See description in section 6.1.
AdUsage	E2	Structure	0..1	See description in section 6.1

Table 17: AdServerPushAds message parameters

Note: The association of the Ad ID and the server to which report the Ad Metrics data for this Ad is taken care by the underlying layer, e.g. HTTP, DCD and BCAST.

The Ad Server MAY be able to pass the parameters as described in the first column of Table 11, and the said parameters SHALL be of the type denoted in the second and third column of Table 11. The cardinality of each parameter in a given request SHALL be as denoted in the fourth column of Table 11.

In contrast, the fifth column in Table 11 is given solely for informative purposes.

Parameter name	XML Type	Data type	Parameter cardinality	Description
AdType	A	Integer	0..1	Indicate if the Ad Content is of type: 0-Default 1-Static 2-Dynamic 3-256 reserved for future use.

Table 18: AdServerPushAds message optional parameters

6.5.2 Delv-1 - Push MobAd Rules

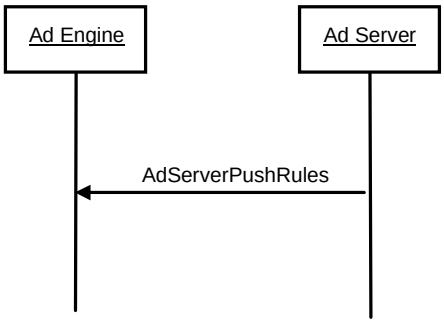

Figure 9.: Flow of Ad Server pushes MobAd Rules to Ad Engine

Message and Information Elements:

Message	Requirement	Direction
AdServerPushRules	Mandatory	Ad Engine ← Ad Server

6.5.2.1 AdServerPushRules message

The "AdServerPushRules" message is used by the Ad Server to push MobAd Rule to MobAd Engine.

In the AdServerPushRules, the Ad Server SHALL be able to pass the parameters as described in the first column of Table 17, and the said parameters SHALL be of the type denoted in the second and third column of Table 17. The cardinality of each parameter in a given message SHALL be as denoted in the fourth column of Table 17.

Message format is as follow:

Parameter name	XML Type	Data type	Parameter cardinality	Description
AdServerPushRules	E		1	Indicate the name of the message
AdServerID	A	String	1	See description in section 6.1.
MobAdRulesInfo	E1	List of structure	1..N	Indicates the MobAdRulesInfo information, for each rule, made available to the Ad Engine. Refer Section 6.4.3.2.1 for MobAdRulesInfo structure.

Table 19: Parameters of Delv-1 AdServerPushRules message

6.5.3 Server-side – Push Notification

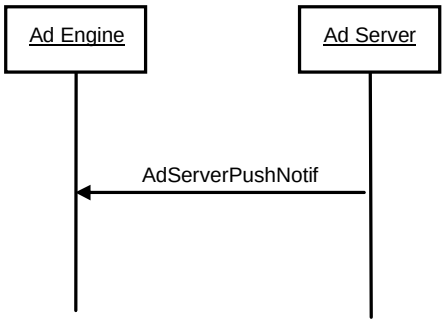

Figure 9: Flow of Ad Server Push Notification to Ad Engine

Message and Information Elements

Message	Requirement	Direction
AdServerPushNotif	Mandatory	Ad Server → Ad Engine

6.5.3.1 AdServerPushNotif message

This message is used by the Ad Server to push notification to Ad Engine.

Message format is as follows:

Parameter name	XML Type	Data type	Parameter cardinality	Description
AdServerPushNotif	E		1	Indicate the name of the message
AdServerID	A	String	1	See description in section 6.1.
NotificationType	A	Integer	1	Indicates the notification types. Detailed type description can be seen in section 5.5.3.1.1.

Table 20: Parameters of Delv-1 AdServerPushNotif message

6.5.3.1.1 NotificationType

The values and the meaning of the notification types are defined in the table below:

Value	Meaning
1	Notification for information that an Ad or campaign needs to be cancelled immediately.
2	Notification for information that MobAd Rules (e.g. SP caching and pre-fetching policies) have changed.
3	Notification to request for Ad Metrics data reporting.
4	Notification to request for new available Ad(s).
5+	Reserved for future use.

Table 21: Notification Type

6.5.4 Broadcast – Ad Server to Ad Engine

Ad Server and Ad Engine MAY support broadcast delivery of Ads and/or Ad Metadata over Delv-1 using modified AdServerPushAds message. The realization of these messages for BCAST adaptation is defined in [MobAd-BCAST Adaptation].

6.6 Common Structures

This sections defines structures that are used in a few of the MobAd interfaces

6.6.1 AdMetricsData

Parameter name	XML Type	Data type	Parameter cardinality	Description
AdID	A	String	1	Refer to description in Section 6.1
MetricType	A	String	0..1	Indicates the type of Metrics being reported. The types are one of the followings: a. Impression b. ClickToWebsite c. ClickToBeContacted d. ClickToLocate e. ClickToCoupon f. ClickToBuy g. ClickToDownload h. ClickToForward i. ClickToRequest j. ClickToDiscard k. ClickToBookmark (save bookmark) l. ClickToCall m. ClickToEmail n. ClickToIM Notes: 1. Support of this list is per the SP policy 2. Identifiers for these and metric type extensions are per the schema set by the SP. Implementation note: The MetricTypes that can be reported back are subject to the capability to support the related interaction type. (cf. section 5.3.3)
MetricValue	A	String	0..1	Indicate the value of the data metric information
ViewDuration	A	Integer	0..1	Length in milli-seconds for which the Ad was viewed
DisplayDateTime	A	String (TimeStamp)	0..1	Date and Time at which the Ad view started

Table 22: AdMetricsData Structure

6.6.2 ContextualData

Note: the XML Type column is an example of the organization of Element with their respective attributes. In the table below all the attributes are child of the ContextualData Element.

Parameter name	XML Type	Data type	Parameter cardinality	Description
ContextualDataID	A	String	0..1, Conditional	See description in section 6.1. Condition: SHALL be provided if more than one context is provided in the request.
AdProvider	A	String	0..1 per requested Ad	Specifies the Ad Provider identifier or brand.
ContextKeyword	E1	Strings	0..N	Identifies all the keywords are meant to describe the Ad opportunity context so as to facilitate the Ad selection.

Table 23: ContextualData Structure

6.6.3 AdContentData

This structure is used in the response messages only.

Note: the Hierarchy column is an example of the organization of Element with their respective attributes

Parameter name	XML type	Data type	Parameter cardinality	Description
MIME-Type	A	String	1	Indicate the mime-media type of the content.
MIME-Version	A	String	0..1, Conditional	The version of the codec specified by the mime-type, if applicable. Condition: This parameter is present if there are multiple versions relevant to the mime-type value (e.g. h263-level1 vs. h263-level2)
AdWidth	A	Integer	0..1, Conditional	Indicates the desired 'width' of the display area in pixels into which the Ad content data should be displayed. Condition: Only applicable for certain MIME-Types
AdHeight	A	Integer	0..1, Conditional	Indicates the desired 'height' of the display area in pixels into which the Ad content data should be displayed. Condition: Only applicable for certain MIME-Types
AdDuration	A	Integer	0..1, Conditional	Indicates the length in milliseconds of media clip if Ad is media (e.g. video or audio) Condition: Only applicable for certain MIME-Types
AdContentSize	A	Integer	1	Indicates the size (in bytes) of the Ad payload
TextEncoding	A	String	1,Conditional	Indicate the encoding scheme used for text, such as ASCI, Unicode,UTF-8. Condition: the MIME-Type is of type text

AdContentPayload	E1	hexBinary	1,Conditional	The actual Ad content data Condition: This parameter is mutually exclusive with AdLocationURI parameter.
AdLocationURI	E1	anyURI	1,Conditional	Indicates the location of the Ad Content. the Ad content data can be retrieved according to the AdLocationURI Condition: This parameter is mutually exclusive with AdContentPayload parameter.
InteractivityAction	E1	List of Structures	0..N	This parameter represents interactive actions associated with this Ad.

Table 24: AdContentData Structure

6.6.4 InteractivityAction

Parameter name	XML Type	Data type	Parameter cardinality	Description
DefaultAction	A	Boolean	0..1	This parameter denotes default action (TRUE)
ActionCode	A	String	1	Contains code for interactivity action associated with an Ad e.g. "ClickToBuy". The code is listed as the following: 1. ClickToWebsite 2. ClickToBeContacted 3. ClickToLocate 4. ClickToCoupon 5. ClickToBuy 6. ClickToDownload 7. ClickToForward 8. ClickToRequest 9. ClickToDiscard 10. ClickToBookmark (save bookmark) 11. ClickToCall 12. ClickToEmail 13. ClickToIM 14 + Reserved for future use Notes: 1. Support of this list is per the SP policy. 2. Identifiers for these and interactivity action code extensions are per the schema set by the SP. Implementation note: The capability to support these interactivity mechanisms normally also depends on the integration of Ad App with the underlying device platform. (cf. section 5.3.3)
ActionURI	A	URI	0..1	Contains action URI associated with the action code specified in ActionCode parameter, e.g. "https://payment.example.com/?3234342".

Table 25: InterActivityAction Structure

6.6.5 AdUsage

Parameter name	XML Type	Data type	Parameter cardinality	Description
DisplayTime	A	String	0..1	Contains information on when the given Ad is to be used, e.g. “Mon-Fri; 22.00-01.00GMT”.
FrequencyCap	A	Integer	0..1	Contains the maximum number of the opportunities where the given Ad is to be used.
PreserveAdFormat	A	Boolean	0..1	When TRUE Ad format MUST be preserved. Default MUST be interpreted as TRUE.
AdValidity	A	String	0..1	The expected period of the lifetime of the Ad, formatted as per RFC 2445

Table 26: InterActivityAction Structure

6.7 Error Descriptions

Errors are handled at the bearer level.

See section 7.3.1.3 for HTTP.

See MobAd over DCD and MobAd over BCAST adaptation specification for error handling in BCAST and DCD delivery cases.

7. Transport Protocol Bindings

7.1 Overview

For MobAd-1 interface, it is left to the device implementation to implement and the transport binding is not specified in the MobAd specifications.

For MobAd-2 interface, the specification defines at least HTTP 1.1 [RFC2616] that Ad Server SHALL support.

For MobAd-3 interface, the specification defines HTTP 1.1 [RFC2616] that both Ad Server and Ad Engine SHALL support.

For Delv-1 interface, the specification defines at least WAP Push, OMA DCD and OMA BCAST, any of which the Ad Server and the Ad Engine MAY support.

7.2 WAP Push Binding

WAP Push enabler [WAP-PUSH] MAY be used for Delv-1 interface messages.

The Ad Server MAY support the delivery of Ads over the Push Access Protocol (PAP) [WAP-PUSH] via the Delv-1 interface.

The Ad Engine MAY support the delivery of Ads over the Push Access Protocol (PAP) [WAP-PUSH] via the Delv-1 interface.

If WAP Push is supported, the Ad Server SHALL include the Push Application ID header – "X-Wap-Application-Id: *x-wap-application:mobad.ua*" and the following mime types used for various Push protocols.

S. No	Push Protocol	MIME content type
1	Push Access Protocol(PAP)	application/vnd.oma.mobad_1_0

Table 27**: Ad Content MIME types for different PUSH mechanisms**

If WAP Push is supported, the Push Client on the terminal SHALL use the X-Wap-Application-Id value to route the push request to the intended application, i.e. Ad Engine in this case.

7.3 HTTP Binding

7.3.1 General

Ad Server and Ad Engine SHALL support Hypertext Transfer Protocol version 1.1 (HTTP1.1 [RFC2616]) for MobAd-3 interfaces.

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

7.3.1.1 Media Type

Ad Server and Ad Engine SHALL support messages formatted as entity-bodies with the following media types:

- application/vnd.oma.mobad_1_0 media type. The application/vnd.oma.mobad_1_0 media type is used when a single MobAd interface message is included in the HTTP request/response.
- multipart/related media type. The multipart/related media type is used when MobAd interface messages are delivered as several inter-related body parts in a single HTTP request/response.
- multipart/mixed media type. The multipart/mixed media type is used when multiple MobAd interface messages are concatenated in a single HTTP request/response.

The header of a MobAd message consists of header fields which in general consist of a field name and its value. Some of the header fields are common [RFC2822] header fields and some values may be specific to the MobAd messages. Figure 10 depicts the conceptual model and example of the encapsulation.

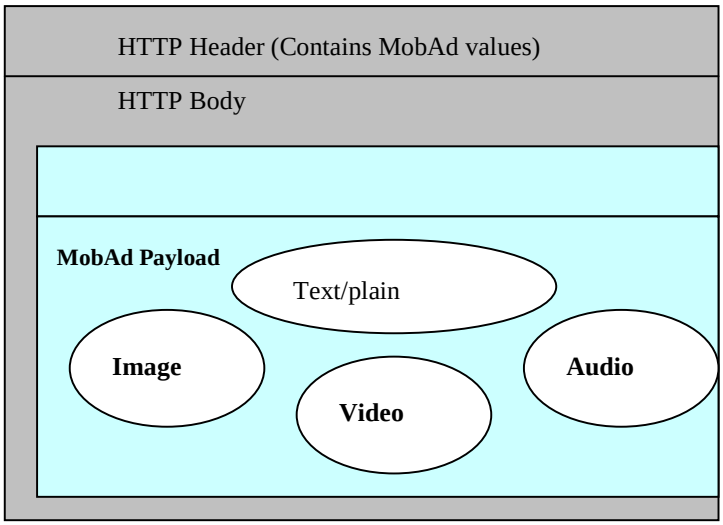

Figure 10: Example of HTTP message containing MobAd payload

7.3.1.2 HTTP Method

Ad Engine SHALL send all MobAd-3 interface messages as HTTP POST method requests.

Ad Engine and SP App MAY send the following optional Headers:

- the Ad Server address in the request line, per the applicable connection profile
- the Host request-header set to the hostname or IP address of the Ad Server, per the applicable connection profile
- the User-Agent request-header set to identify the host device (e.g. “vendor-model/version”), and the name and version of the Ad Engine as user agent initiating the request
- the Accept request-header with value “application/vnd.oma.mobad_1_0, multipart/related, multipart/mixed”
- the Accept-Charset request-header with value per the supported HTTP supported content types character sets(e.g. UTF8)
- 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 X-Wap-Profile request-header set to the URI of the User Agent Profile for the host device
- the Content-Length entity-header set to the length of the entity-body
- the Content-Type entity-header with value “application/vnd.oma.mobad_1_0”, “multipart/related”, or “multipart/mixed”, as applicable
- the MobAd-3 or MobAd-2 message(s) as message-body

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

Ad Server SHALL send all MobAd-3 interface messages as responses to the corresponding Ad Engine requests, and SHALL send all MobAd-2 interface messages as responses to the corresponding SP App requests, 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 Ad Server
- 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/vnd.oma.mobad_1_0", "multipart/related", or "multipart/mixed", as applicable
- the MobAd-3 or MobAd-2 message(s) as message-body, if the transaction is successful

7.3.1.3 Error handling

When there is no MobAd message to send in response to an Ad Engine request or a SP App request, the Ad Server SHALL send a 204 No Content response. Other allowed status codes, reflecting the outcome of the HTTP POST request, are defined in [RFC2616]. The Status-Line header MUST be included in the response to reflect the outcome of the HTTP POST submission.

7.3.2 MobAd-3 Interface Message Bindings

The table below gives an overview of how MobAd-3 interface messages are bound to the HTTP based protocol stack.

Message	Ad Engine ←→ Ad Server	HTTP Method
AdEngineAdRequest	→	HTTP POST
AdEngineAdResponse	←	HTTP Response (including 200 OK of the underlying method)
AdEngineMetricsReport	→	HTTP POST :Ad Server responds with 200 OK of the underlying method
AdEngineRulesRequest	→	HTTP POST
AdEngineRulesResponse	←	HTTP Response (including 200 OK of the underlying method)

Table 28: MobAd-3 Interface Message Bindings When Uses HTTP Based Stack

7.3.3 MobAd-2 Interface Message Bindings

If HTTP is supported over MobAd-2 interface, the table below gives an overview of how MobAd-2 interface messages are bound to the HTTP based protocol stack:

Message	SP App ←→ Ad Server	HTTP Method
SPAppAdRequest	→	HTTP POST
SPAppAdResponse	←	HTTP Response (including 200 OK of the underlying method)
SPAppMetricsReport	→	HTTP POST :Ad Server responds with 200 OK of the

		underlying method

Table 29: MobAd-2 Interface Message Bindings When Uses HTTP Based Stack

7.4 DCD Adaptation

Ad Server and Ad Engine MAY use OMA DCD enabler for the Delv-1 interface.

MobAd entities operating over the OMA DCD enabler SHALL comply with [MOBAD-TS-DCD] in addition to this specification.

7.5 BCAST Adaptation

Ad Server and Ad Engine MAY use OMA BCAST enabler for the Delv-1 interface. MobAd entities operating over the OMA BCAST enabler SHALL comply with [MOBAD-TS-BCAST] in addition to this specification.

8. Security Considerations

8.1 Ad Engine – Ad Server Authentication and Communications

Ad Engine and Ad Server authentication is based on the mechanism defined in [RFC2617]. Implementations conforming to this specification MUST support this mechanism for "Basic" and "Digest Access Authentication". The Base64 character encoded "Basic" and "MD5" algorithm of the "Digest Access Authentication" authentication schemes MAY be supported. The HTTP headers and parameters that MUST be supported are described in the previous sections for request and response headers.

Ad Server implementations conforming to this specification MAY support both the ability to challenge unauthenticated requests and also accept authentication request headers in a request; which will not require subsequent challenge responses unless some part of the credential is incorrect. The latter requirement is required to address the need for minimal request/response traffic for mobile networks.

The authentication mechanisms defined by [RFC2617] address the protection of the authentication credentials. However, the remainder of the HTTP request and response messages are available to the eavesdropper. For more robust security for the HTTP connection, TLS 1.1 [RFC4346], TLS 1.2 [RFC5246], HTTPS, or some form of upgrading to TLS over HTTP [RFC2817] [RFC2818] SHOULD be used.

When operating over HTTP:

- The Ad Server SHALL support TLS 1.1 [RFC4346],
- The Ad Server SHOULD support TLS 1.2 [RFC5246]
- The Ad Engine SHALL support TLS 1.1 or TLS 1.2
- The Ad Engine SHALL identify which TLS version the Ad Server is using and set-up the communication accordingly.
- The Ad Server MUST support all of the following cipher suites, all of which provide authentication, confidentiality and integrity, when using a TLS 1.1 session:
    - TLS_RSA_WITH_AES_128_CBC_SHA256
    - TLS_DH_DSS_WITH_AES_128_CBC_SHA256
    - TLS_DH_RSA_WITH_AES_128_CBC_SHA256
    - TLS_DHE_DSS_WITH_AES_128_CBC_SHA256
    - TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
    - TLS_RSA_WITH_AES_128_GCM_SHA256
    - TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
    - TLS_DH_RSA_WITH_AES_128_GCM_SHA256
    - TLS_DHE_DSS_WITH_AES_128_GCM_SHA256
    - TLS_DH_DSS_WITH_AES_128_GCM_SHA256
    - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
    - TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
    - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
    - TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256

- The Ad Engine MUST support at least one of the above cipher suites when using a TLS 1.1 session

8.2 SP App to Ad Server Communications

SP App and Ad Server components reside both in the Service Provider domain. As a consequence the mechanisms that may be required to secure (e.g. Authentication, Anonymisation, Integrity, Encryption) communication between these two components are set by SP Policies, are deployment choices and are out of scope of this specification.

8.3 Ad App to Ad Engine Communications

Ad App and Ad Engine components reside both on the same device. As a consequence mechanisms that may be required to secure communication between these two components (e.g. Integrity, Authentication, and Encryption) rely on the underlying platform implementation and are out of scope of this specification.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-MobAd_Core-V1_0	20 Mar 2012	Status changed to Approved by TP: OMA-TP-2012-0116-INP_MobAd_V1_0_ERP_for_Final_Approval

Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for Ad Engine

Item	Function	Reference	Requirement
MOBAD_CORE-E-001-M	Support receiving AdAppAdRequest message from Ad App	Section 6.2.1.1	
MOBAD_CORE-E-002-M	Support sending AdAppAdResponse message to Ad App	Section 6.2.1.2	
MOBAD_CORE-E-003-M	Support receiving AdAppMetricsReport message from Ad App	Section 6.2.2.1	
MOBAD_CORE-E-004-M	Support sending AdEngineAdRequest message to Ad Server	Section 6.4.1.1	http1.1, TLS 1.1 or TLS 1.2
MOBAD_CORE-E-005-M	Support receiving AdEngineAdResponse message from Ad Server	Section 6.4.1.2	http1.1, TLS 1.1 or TLS 1.2
MOBAD_CORE-E-006-M	Support sending AdEngineMetricsReport to the Ad Server	Section 6.4.2.1	http1.1, TLS 1.1 or TLS 1.2
MOBAD_CORE-E-007-M	Support prefetching, caching and managing Ads from the Ad Server	Section 5.1.2	
MOBAD_CORE-E-008-M	Support selecting Ads for an Ad App	Section 5.1.4	
MOBAD_CORE-E-009-M	Support Ad Metrics Data Aggregation	Section 5.1.6	
MOBAD_CORE-E-010-M	Support Applying Rules and Policies	Section 5.1.7	MOBAD_CORE-E-014-O or (MOBAD_CORE-E-011-O and MOBAD_CORE-E-012-O)
MOBAD_CORE-E-011-O	Support sending AdEngineRulesRequest message to Ad Server	Section 6.4.3.1	http1.1, TLS 1.1 or TLS 1.2
MOBAD_CORE-E-012-O	Support receiving of AdEngineRulesResponse from Ad Server	Section 6.4.3.2	http1.1, TLS 1.1 or TLS 1.2
MOBAD_CORE-E-013-O	Support receiving AdServerPushAds message from Ad Server	Section 6.5.1.1	WAP-PUSH or OMA DCD or OMA BCAST
MOBAD_CORE-E-014-O	Support receiving AdServerPushRules message from Ad Server	Section 6.5.2.1	WAP-PUSH or OMA DCD or OMA BCAST
MOBAD_CORE-E-015-O	Support receiving AdServerPushNotif message from Ad Server Note: Requirement column applies to NotificationType related to MobAd Rules.	Section 6.5.3.1	WAP-PUSH or OMA DCD or OMA BCAST MOBAD_COR

Item	Function	Reference	Requirement
			E-E-011-O and MOBAD_CORE-E-012-O
MOBAD_CORE-E-016-O	Support Ad Metrics Data Validation	Section 5.1.6	
MOBAD_CORE-E-017-O	Support suspend and resume the MobAd functions	Section 5.1.10	

B.2 SCR for Ad Server

Item	Function	Reference	Requirement
MOBAD_CORE-S-001-M	Support receiving SPAppAdRequest message from SP App	Section 6.3.1.1	http 1.1
MOBAD_CORE-S-002-M	Support sending SPAppAdResponse message to SP App	Section 6.3.1.2	http 1.1
MOBAD_CORE-S-003-M	Support receiving SPAppMetricsReport message from SP App	Section 6.3.2.1	http 1.1
MOBAD_CORE-S-004-M	Support receiving AdEngineAdRequest message from Ad Engine	Section 6.4.1.1	http1.1, TLS 1.1 or TLS 1.2
MOBAD_CORE-S-005-M	Support sending AdEngineAdResponse message to Ad Engine	Section 6.4.1.2	http1.1, TLS 1.1 or TLS 1.2
MOBAD_CORE-S-006-M	Support receiving AdEngineMetricsReport from Ad Engine	Section 6.4.2.1	http1.1, TLS 1.1 or TLS 1.2
MOBAD_CORE-S-007-M	Support selecting Ads for an Ad Engine and SP App	Section 5.2.1.1	
MOBAD_CORE-S-008-M	Support delete and/or replace of Ads	Section 5.2.1.2	
MOBAD_CORE-S-009-O	Support receiving AdEngineRulesRequest message from Ad Engine	Section 6.4.3.1	http1.1, TLS 1.1 or TLS 1.2
MOBAD_CORE-S-010-O	Support sending AdEngineRulesResponse to Ad Engine	Section 6.4.3.2	http1.1, TLS 1.1 or TLS 1.2
MOBAD_CORE-S-011-O	Support sending AdServerPushAds message to Ad Engine	Section 6.5.1.1	WAP-PUSH or OMA DCD or OMA BCAST
MOBAD_CORE-S-012-O	Support sending AdServerPushRules message to Ad Engine	Section 6.5.2.1	WAP-PUSH or OMA DCD or OMA BCAST
MOBAD_CORE-S-013-O	Support sending AdServerPushNotif message to Ad Engine	Section 6.5.3.1	WAP-PUSH or OMA DCD or OMA BCAST