MobAd Technical Specification

- DCD Adaptation

Approved Version 1.0 – 20 Mar 2012

Open Mobile Alliance

OMA-TS-MobAd_DCD_Adaptation-V1_0-20120320-A

Contents

Figures

1. Scope

This document describes the additional functionalities necessary to enable MobAd services by using and leveraging the DCD Enabler.

These functionalities comprise at least the Registration and Deregistration of the Ad Engine to the DCD enabler, Channel Subscription and Unsubscription, the usage of Application Profile(s) and Subscription Filter(s) to personalize and contextualize the Ad channel delivery.

This document defines the mapping of [MobAd-Core-TS] messages to [DCD-TSS] messages and specifies the usage of administrative messages over additional interfaces that have no equivalent in the [MobAd-Core-TS].

This document supplements the Core MobAd technical specification.

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/

[DCD-AD] "Dynamic Content Delivery Architecture Document", Version 1.0, Open Mobile Alliance™, OMA-AD-DCD-V1_0-20081223-C, URL:http://www.openmobilealliance.org/

[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/

[MobAd-AD] "Mobile Advertising Architecture Document", Version 1.0, Open Mobile Alliance™, OMA-AD-MobAd-V1_0, URL:http://www.openmobilealliance.org/

[MobAd-Core-TS] "Mobile Advertising Core Technical Specification", Version 1.0, Open Mobile Alliance™, OMA-TS-MobAd-V1_0, URL:http://www.openmobilealliance.org/

2.2 Informative References

[OMADICT] "Dictionary for OMA Specifications", Version x.y, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_7, URL:http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

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

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

3.2 Definitions

Ad App	See [MobAd-Core-TS]
Ad Engine	See [MobAd-Core-TS]
Ad Server	See [MobAd-Core-TS]
Application Profile	See [DCD-TSS]
Channel Guide	See [DCD-TSS]
Channel Metadata	See [DCD-TSS]
DCD Channel	See [DCD-TSS]
DCD Client	See [DCD-TSS]
DCD Server	See [DCD-TSS]

3.3 Abbreviations

AP	Application Profile
DCD	Dynamic Content Delivery
DCD-CADE	Dynamic Content Delivery – Data Exchange with Client Application
DCD-CAR	Dynamic Content Delivery – Client Application Registration
DCD-CPDE	Dynamic Content Delivery – Data Exchange with DCD Content Provider
DCD-CPR	Dynamic Content Delivery – Content Provider Registration
DECA	DCD Enabled Client Application
MobAd	Mobile Advertising
OMA	Open Mobile Alliance

4. Introduction

The OMA Mobile Advertising (MobAd) 1.0 enabler provides a set of functions and interfaces designed to enable contextualisation and personalization of advertisements, mechanisms to support interactive advertisements, advertising metrics for recording user's behaviour and the impact of mobile advertising.

The OMA Dynamic Content Delivery (DCD) 1.0 enabler provides a set of functions and interfaces designed to enable subscription, personalization, and delivery of content to mobile users on a one-to-one (point-to-point) or one-to-many (broadcast) basis. The content delivery functionality of the DCD enabler is designed to support multiple underlying network technologies (e.g. different network types and/or bearers).

This document describes the interactions between the MobAd enabler and the DCD enabler related to Mobile Advertising operations over the DCD.

4.1 Version 1.0

Version 1.0 will specify all functions, behaviors and data (e.g.: metadata) necessary to enable MobAd V1_0 operations using and leveraging DCD.

5. Architecture

5.1 Architecture diagram

The following architecture diagram shows the use of DCD enabler by MobAd enabler components:

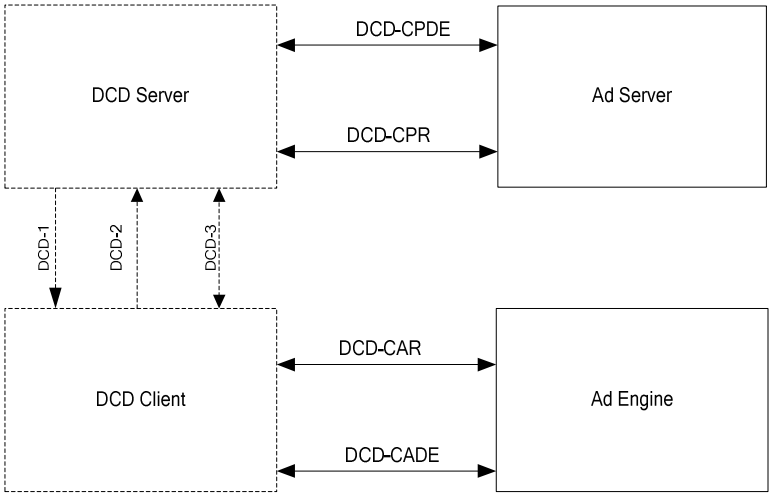

Figure 1 : Architectural view of MobAd enabler and DCD enabler.

Legend:

Dash line - interfaces and components out of scope for this specification

Solid line - components of MobAd enabler and DCD interfaces used by MobAd enabler

The representation of bidirectional arrows is consistent with the description in the [DCD AD].

The Ad Server acts as the DCD Content Provider as described in [DCD-TSS].

The Ad Server uses the DCD-CPDE interface to the DCD Server for delivery of Ads, Ad references, Ad metadata, and Ad Metrics and Ad request. The Ad Server uses the DCD-CPR interface to the DCD Server for Ad channel registration, subscription for Ad channels, update of Ad channel metadata, and reporting of Ad channel usage. The operations over DCD-CDPE and CDC-CPR are specified in [DCD-TSS].

Note: An Ad Channel in this specification corresponds to a Channel in [DCD-TSS]

The Ad Engine acts as the DCD Enabled Client Application (DECA) as described in [DCD-TSS].

The Ad Engine uses the DCD-CADE interface to the DCD Client to delivery Ads, Ad references, and Ad metadata, Ad Metrics and Ad request, to discover available Ad channels, and to suspend/resume Ad channels. The Ad Engine uses the DCD-CAR interface to the DCD Client for Ad Engine registration, subscription for Ad channels, and update of metadata for Ad channel personalization. The operations over DCD-CADE and CDC-CAR are specified in [DCD-TSS].

MobAd interfaces DELV-01 and MobAd-3 described in section 5.3.3 of [MobAd-AD] are mapped to the DCD-CAR, DCD-CADE, DCD-CPR and DCD-CPDE interfaces specified in [DCD-TSS].

5.2 MobAd over DCD high level end to end flow

This end-to-end flow illustrates the operations performed by the MobAd enabler to operate with the DCD enabler.

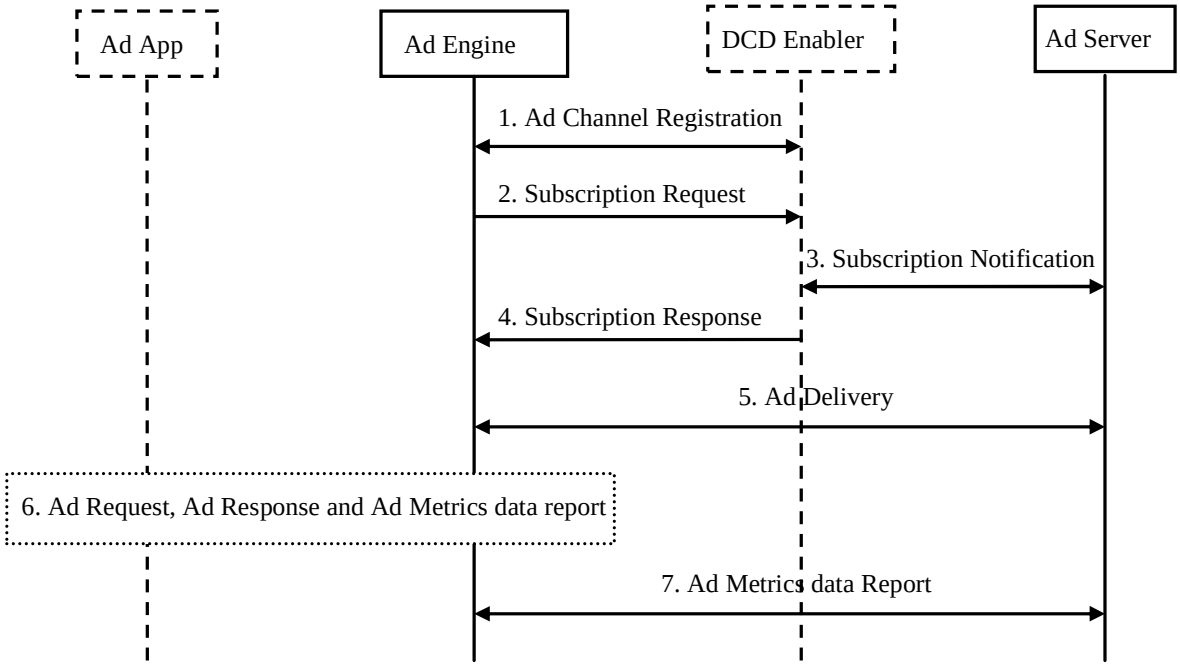

Figure 2: MobAd over DCD end to end flow.

Prerequisite: Ad Server is registered with the DCD Enabler as a DCD Content Provider

1: The Ad Engine registers to the DCD Enabler by Providing an Application Profile. Upon registration Ad Engine receives the list of applicable Ad-related DCD Channels.

2: The Ad Engine requests subscription to selected Ad-related DCD Channels (e.g. according to user's interests/preferences/Ad App) or it is auto-subscribed based on the Ad Server preferences specified in the Channel Metadata

3: The DCD Enabler notifies the Ad Server about the Ad Engine subscriptions

4: The Ad Engine gets the response for the subscriptions

5: the Ad Server provides Ads to the DCD Enabler for the channels subscribed by the Ad Engine (pushed to Ad Engine or returned upon Ad Engine request)

6: This step covers the Ad Request, Ad response and the Ad Metrics data reporting operation between the Ad App and the Ad Engine that are described in the [MobAd-Core-TS]

7: The Ad Engine reports Ad Metrics data to the Ad Server via the DCD enabler using DCD content submission transaction. These metrics are opaque to the DCD Enabler.

6. MobAd over DCD Operations

6.1 Registration and Deregistration

6.1.1 Ad Engine Registration

The Ad Engine SHALL register with the DCD Enabler over DCD-CAR interface to be used for Ad delivery via the DCD-CADE interface. The Ad Engine SHALL register with the DCD Client using the DCD-CAR ApplicationRegistrationRequest message as described in section 7.3.1.1 of [DCD-TSS]. The Ad Engine SHALL supply the DCD Application Profile for the registration, which is described in section 8.1.2 of [DCD-TSS]. Depending on the Service Provider policy, Ad Engine registers once by providing consolidated Application Profile with combined or common preferences for all Ad Applications, or multiple times providing Application Profiles with preferences per a single Ad Application or a group of Ad Applications (see section 7 “Application Profile” for details).

When the Ad Engine receives a DCD-CAR ApplicationRegistrationResponse message with the Channel Guide in the response, the Ad Engine can selectively subscribe to the Ad-related DCD Channel to obtain the advertisement through the DCD Enabler.

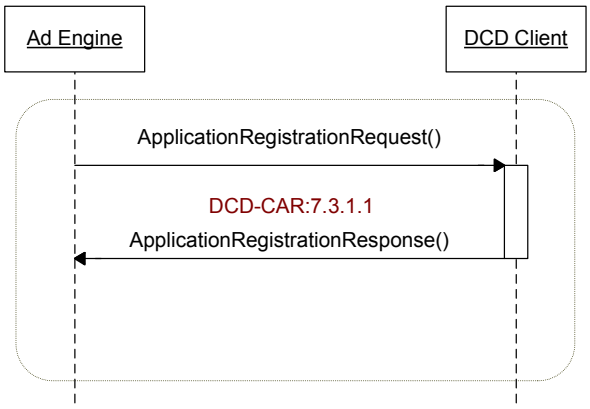

Figure 3Figure 2: MobAd over DCD end to end flow.

6.1.2 Ad Engine Deregistration

The Ad Engine can terminate delivery of Ad over DCD at any point during the DCD Session. The decision to terminate the Ad delivery over DCD will be made by the Ad Engine based on Service Provider provisioning or other mechanisms and is considered out of scope for this specification.

In the case of single registration, the Ad Engine deregisters with the DCD Enabler to terminate Ad delivery to all Ad Applications. In the case of multiple registrations, the Ad Engine deregisters with the DCD Enabler to terminate Ad delivery only to the Ad Applications covered by Application Profile associated with this particular deregistration. To deregister with the DCD Enabler, the Ad Engine SHALL deregister with the DCD Client using the DCD-CAR ApplicationDeregistrationRequest message as described in section 7.3.1.2 of [DCD-TSS]. At deregistration, the Ad Engine SHALL supply the ‘Application-ID’ element from the appropriate DCD Application Profile.

When the Ad Engine receives a DCD-CAR ApplicationDeregistrationResponse message, the Ad Engine is disconnected from the DCD Enabler.

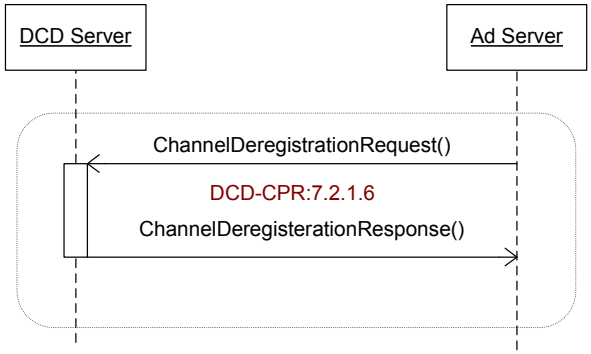

Figure 4Figure 2: MobAd over DCD end to end flow.

6.1.3 Ad Server Channel Registration

To register Ad channels with the DCD Enabler the Ad Server SHALL send a DCD-CPR ChannelRegistrationRequest message as described in section 7.2.1.5 of [DCD-TSS] to the DCD Server.

Ad Server SHALL provide ‘Channel-Metadata’ parameters to facilitate adding these Ad channels to the Channel Guide by the DCD Server, this Channel Guide can be sent by the DCD Server to the Ad Engine through the DCD Client during the Ad Engine registration process. Channel Guide is the list of Channel Metadata for the Ad channels matching Ad Engine preferences.

The Ad Server SHALL set the value for the sub element ‘general-channel-metadata’ of ‘Channel-Metadata’ parameter to describe the channel (e.g. channel-id, channel-name, updated) as described in section 8.2.2.2.1 of [DCD-TSS].

The Ad Server SHALL set the value for the sub element ‘channel-publication-metadata’ of ‘Channel-Metadata’ parameter to specify the publication schedule depending on its delivery method as described in section 8.2.2.2.4 of [DCD-TSS]. The Ad Server SHALL set the value for the element ‘notify-on-subscription-change’ as “true” to get the notification when the Ad Engine subscribes to or unsubscribes from this channel. The ‘notify-on-subscription-change’ element is an indication that Ad Server needs to be informed when the Ad Engine subscribes to the channel, or changes an existing subscription as described in section 8.2.2.2.4 [DCD-TSS].

To facilitate matching of Ad channels with Ad Engine and associated Ad Applications, the Ad Server SHALL specify at least one of the following when registering channels with DCD Server:

- the values for the mime-types and content-types parameters of general-channel-metadata subset of Channel Metadata as described in section 8.2.2.2.1 of [DCD-TSS]. The mime-types attribute is a comma-separated list of MIME types for content items in the offered channel (e.g. “application/atom+xml”). The content-types attribute is a comma-separated list of strings that describe the type of channel content e.g. by “category”, “tag” or "relation" (e.g. “advertisement”).
- the value for the matching-applications parameter of general-channel-metadata subset of Channel Metadata as described in section 8.2.2.2.1 of [DCD-TSS]. The matching-applications attribute is a comma separated list of Application IDs (e.g. “//example.com/mobad/adengine/v.1.0”, “//example.com/mobad/adengine/v.1.0/ad-app/video”) of the DCD Enabled Client Applications matching registered DCD Channel for Ad delivery.

Upon receiving the DCD-CPR ChannelRegistrationResponse message as described in section 7.2.1.5 of [DCD-TSS] from the DCD Server, the Ad Server can send the advertisement through the DCD Channel.

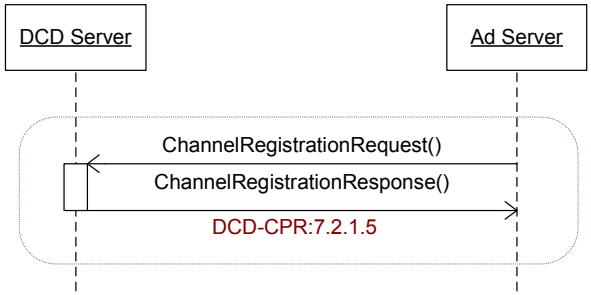

Figure 5Figure 2: MobAd over DCD end to end flow.

6.1.4 Ad Server Channel Deregistration

To deregister a channel with the DCD Enabler the Ad Server sends a DCD-CPR ChannelDeregistrationRequest message to the DCD Server as described in section 7.2.1.6 of [DCD-TSS]. The Ad Server SHALL set the value for 'Channel-ID' parameter to specify the channel to be deregistered.

Upon receiving the DCD-CPR ChannelDeregistrationResponse message from the DCD Server, the Ad Server SHALL no longer deliver advertisement content for the deregistered DCD Channel.

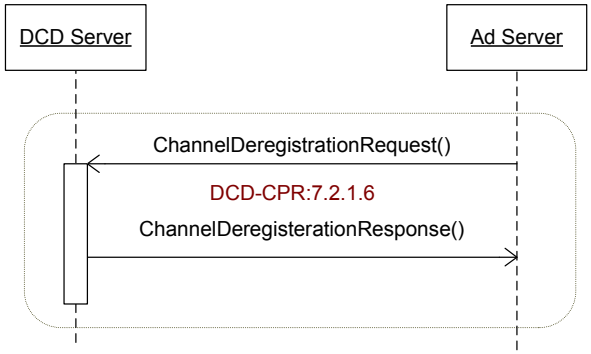

Figure 6Figure 2: MobAd over DCD end to end flow.

6.2 Subscription and Unsubscription

6.2.1 Ad Server initiated Channel Subscription

The Ad Server MAY initiate Ad Engine subscription to the registered advertising channel by specifying the value for the subscription-required attribute of general-channel-metadata subset of Channel Metadata as described in section 8.2.2.2.1 of [DCD-TSS]. If the value of subscription-required is "True", the DCD Client SHALL auto-subscribe appropriate application to this channel. As described in section 6.1.3, the DCD Client can select Ad Engine as an appropriate DCD Enabled Client

Application for this channel based on the Channel Metadata value for the matching-applications parameter or values for mime-type and content-type parameters.

Additionally, the Ad Server can target particular device/subscriber with Ad Engine subscription to the registered advertising channel by using external subscription mechanism, as described in section 5.5 of [DCD-TSS]. In this scenario, the Ad Server SHALL send the DCD-CPR SubscriptionNotification message to the DCD Server as described in section 7.2.1.3 of [DCD-TSS]. This notification message contains mandatory elements 'Delivery-Endpoint-Info' and 'Channel-Metadata'. To facilitate automatic subscription, 'Delivery-Endpoint-Info' SHALL specify the target device/subscriber and SHOULD specify Ad Engine as a target application. The element 'Channel-Metadata' SHALL include the identifier of the advertising channel.

Upon the subscription notification from the Ad Server, the DCD Client SHALL send the DCD-CAR SubscriptionValidationRequest message to the Ad Engine as described in section 7.3.1.7 of [DCD-TSS]. Consecutively, the Ad Engine SHALL send the DCD-CAR SubscriptionValidationResponse message to the DCD Client. Depending on advertising service provider policy, the Ad Engine SHOULD be required to accept the subscription.

Upon receiving DCD-CAR SubscriptionValidationResponse message from the Ad Engine, the DCD Server SHALL send the DCD-CPR SubscriptionNotificationResponse message to the Ad Server as described in section 7.2.1.3 of [DCD-TSS].

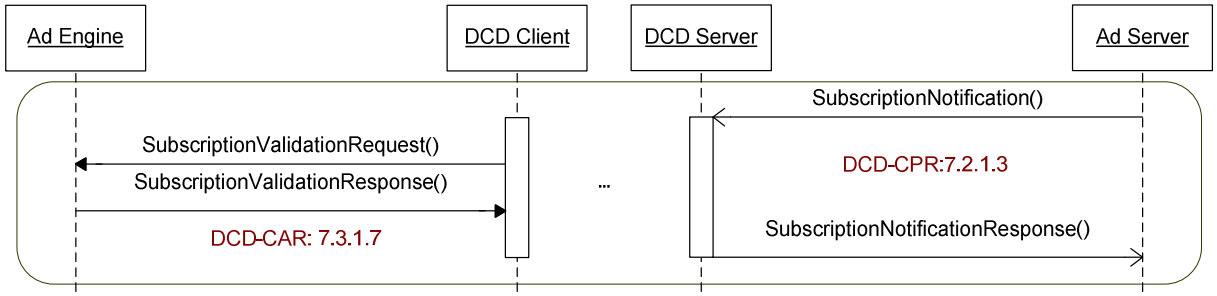

Figure 7Figure 2: MobAd over DCD end to end flow.

6.2.2 Ad Engine initiated Channel Subscription

Upon receiving a DCD-CAR ApplicationRegistrationResponse message including 'Channel-discovery-information' element, the Ad Engine can retrieve the Channel Guide, identify Ad channels, and subscribe to these channels using DCD Enabler.

If the DCD-CAR ApplicationRegistrationResponse message does not contain 'Channel-discovery-information' element, the Ad Engine SHOULD send a DCD-CADE ChannelDiscoveryRequest message to the DCD Client as described in section 7.3.2.1 of [DCD-TSS]. When the Ad Engine receives a DCD-CADE ChannelDiscoveryResponse message, the Ad Engine can retrieve the Channel Guide, identify Ad channels, and subscribe to these channels using DCD Enabler.

If a new Ad channel is registered by the Ad Server, the Ad Engine receives the DCD-CADE ChannelDiscoveryInfo message including the element 'Channels-added' for the new channel as described in section 7.3.2.5 of [DCD-TSS]. This ChannelDiscoveryInfo message is sent from the DCD Client to the Ad Engine to communicate availability of new channels and removal of or updates to the existing channels. Based on this channel discovery information, the Ad Engine can subscribe to the new or updated channel for the Ad delivery over DCD.

The Ad Engine SHALL subscribe to selected Ad channel by sending a DCD-CAR SubscriptionRequest message including 'Channel-Metadata' element to the DCD Client as described in section 7.3.1.3 of [DCD-TSS]. The Ad Engine SHALL set the value for the sub element 'delivery-personalization-metadata' of 'Channel-Metadata' (e.g. 'channel-id') as described in section 8.2.2.1.2 of [DCD-TSS].

To get notified by the DCD Client upon Ad arrival, the Ad Engine SHALL set value of Channel Metadata 'content-availability-notification' element as "true". The 'content-availability-notification' element is an indication that the Ad Engine should be given explicit notification from the DCD Client when new content has been received for this channel as described in section 8.2.2.1.2 [DCD-TSS].

Upon the subscription request from the Ad Engine, the DCD Server SHALL send DCD-CPR SubscriptionRequest message to the Ad Server as described in section 7.2.1.1 of [DCD-TSS]. Based on this message, the Ad Server realizes that the Ad Engine requests subscription for the Ad channel. The Ad Server SHALL send DCD-CPR SubscriptionResponse message to the DCD Server to complete the channel subscription process.

When the Ad Engine receives the DCD-CAR SubscriptionResponse message from the DCD Client, the Ad Engine can start receiving advertisement over the subscribed channel.

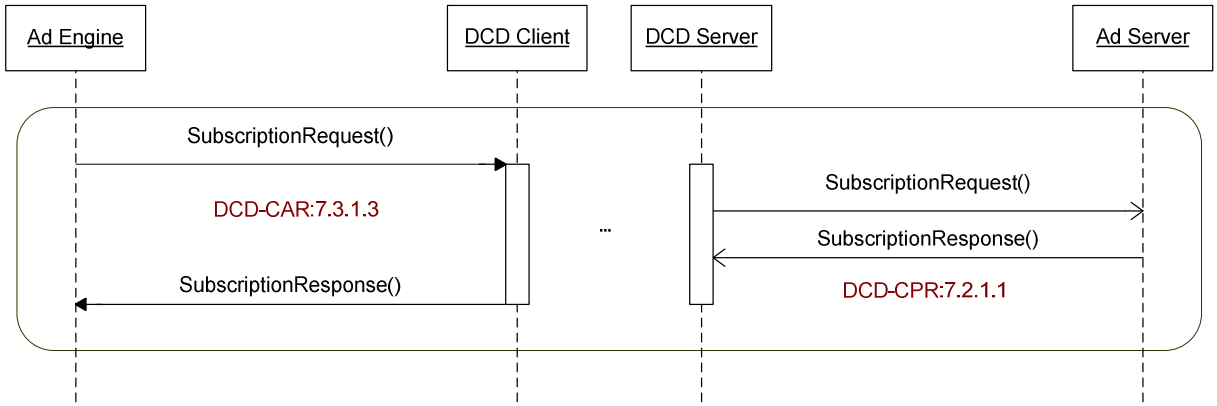

Figure 8Figure 2: MobAd over DCD end to end flow.

6.2.3 Ad Server initiated Channel Deregistration

When the Ad Server deregisters the Ad Channel which was subscribed by Ad Engines, the Ad Server SHALL send DCD-CPR ChannelDeregistrationRequest message to the DCD Server as described in section 7.2.1.6 of [DCD-TSS]. Following the interaction between the DCD Server and DCD Client, the Ad Engines SHALL receive the DCD-CAR UnsubscriptionNotification message from the DCD Client as described in section 7.3.1.6 of [DCD-TSS]. With this message, the Ad Engine is informed that the advertisement delivery over that Ad channel is no longer available. Finally, the Ad Server SHALL receive the DCD-CPR ChannelDeregistrationResponse message from the DCD Server.

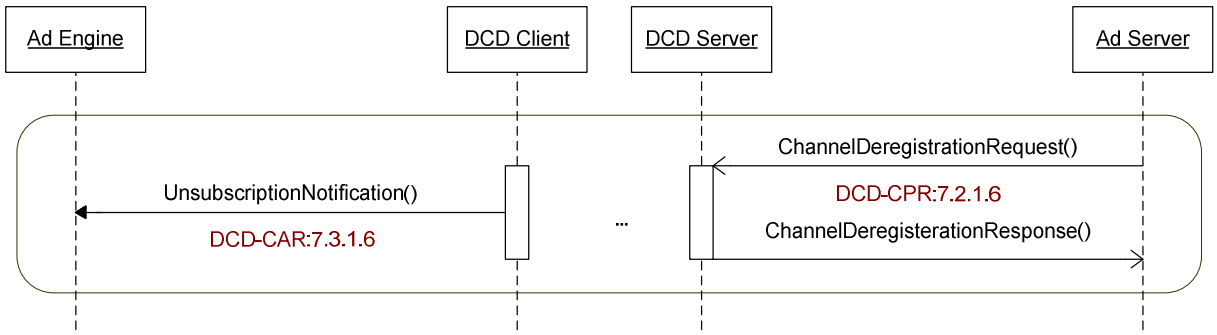

Figure 9Figure 2: MobAd over DCD end to end flow.

6.2.4 Ad Engine initiated Channel Unsubscription

If the Ad Engine decides to terminate delivery of advertising over an Ad channel, the Ad Engine SHALL unsubscribe the Ad channel sending a DCD-CAR UnsubscriptionRequest message including 'Channel-ID' element to the DCD Client as described in section 7.3.1.5 of [DCD-TSS].

Upon the unsubscription request from the Ad Engine, the Ad Server SHALL receive the DCD-CPR UnsubsciptionNotification message from the DCD Server as described in section 7.2.1.2 of [DCD-TSS] and the DCD Client SHALL send the DCD-CAR UnsubscriptionResponse message to the Ad Engine as described in section 7.3.1.5 of [DCD-TSS].

When the Ad Engine receives the DCD-CAR UnsubscriptionResponse message from the DCD Client, the Ad Engine can no longer receive advertisement over this Ad channel.

If the Ad Engine deregisters with the DCD Client and it still has active subscriptions to the Ad channels, the Ad Server SHALL receive the DCD-CPR UnsubscriptioinNotification message from the DCD Server including the information regarding the unsubscribed Ad channels (i.e. the DCD Client terminates all subscriptions of deregistered Ad Engine).

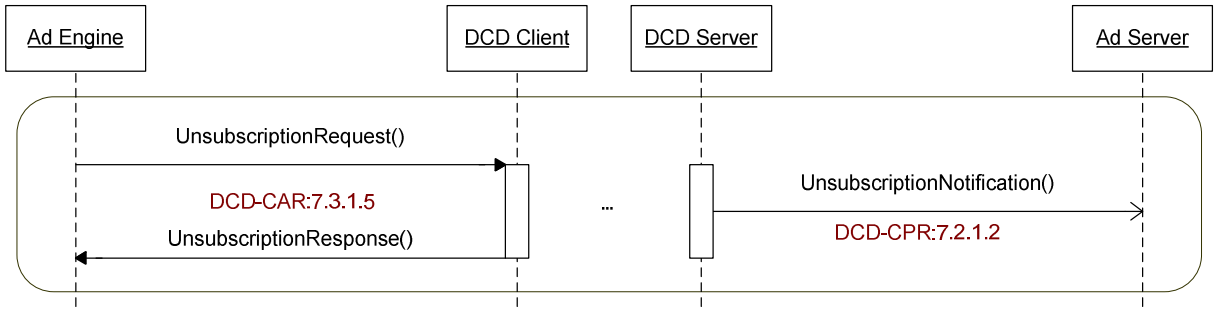

Figure 10Figure 2: MobAd over DCD end to end flow.

6.2.5 Ad Server initiated Channel Update

To update the Channel Metadata for the registered Ad channel, the Ad Server SHOULD send the DCD-CPR ChannelRegistrationRequest message to the DCD Server including channel identifier and modified elements for the Channel Metadata as described in section 7.2.1.5 of [DCD-TSS]. If the Ad Engine subscribed that modified Ad channel, the DCD Server passes it to the DCD Client and the DCD Client SHOULD send the DCD-CAR ChannelMetadataUpdate message to the Ad Engine as described in section 7.3.1.8 of [DCD-TSS]. Finally, the DCD Server SHOULD send the DCD-CPR ChannelRegistrationReponse message to the Ad Server.

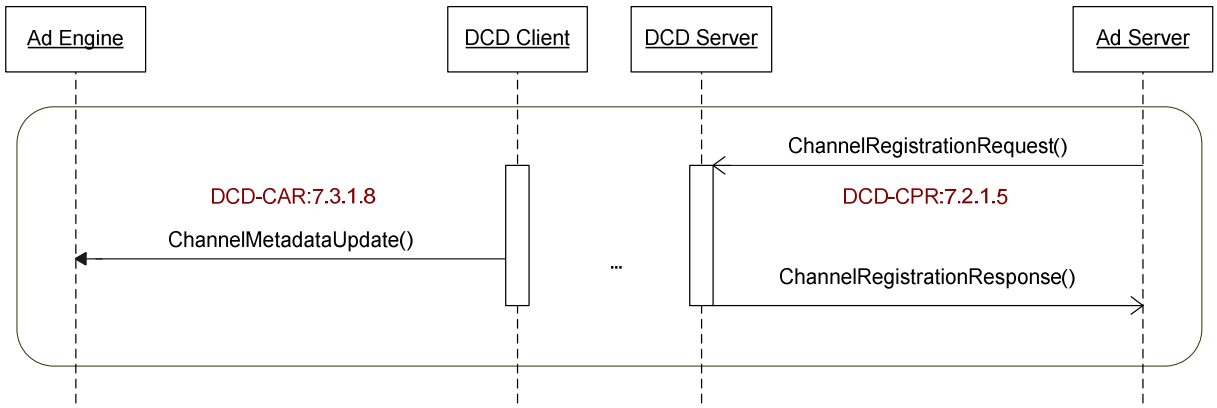

Figure 11Figure 2: MobAd over DCD end to end flow.

6.2.6 Ad Engine initiated Subscription Personalization

To facilitate targeted advertising according to the preferences of installed Ad Applications and/or device user advertising preferences, the Ad Engine can customize its subscriptions using DCD subscription personalization mechanism as described in section 5.5.1.2 of [DCD-TSS]. To customize channel subscription, the Ad Engine SHOULD provide subscription filters to the Ad Server upon subscribing to an Ad channel via the DCD Client. Ad Engine SHOULD either provide multiple subscription filters representing preferences of particular Ad Applications or a single subscription filter representing consolidated preferences. Ad Server SHOULD issue subscription-id for each filter and SHOULD send DCD-CPR SubscriptionUpdate message to DCD Server as described in section 7.2.1.4 of [DCD-TSS]. Upon receiving this message, the DCD Server adds device user to the group of subscribers for Ad channel associated with this subscription-id.

Upon this subscription personalization, the DCD Server can deliver this advertisement content to the group of subscribers associated with the subscription identifiers in the DCD-CPDE ContentUpdate or the DCD-CPDE ContentUpdateResponse message. When Ad Engine receives Ad content relevant to provided subscription filters it can either send it directly to appropriate Ad Applications or provide it upon Ad Application Ad request. When subscription filters represent preferences of particular Ad Applications, the Ad Engine can use mapping of subscription filters to the applications to facilitate Ad delivery.

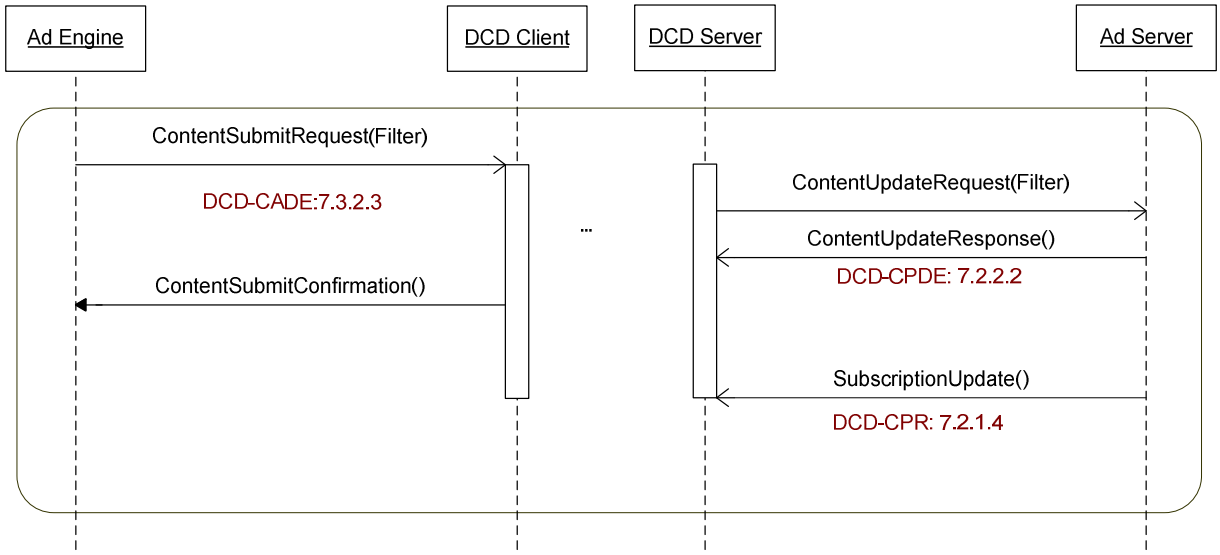

Figure 12Figure 2: MobAd over DCD end to end flow.

6.3 Delivery

6.3.1 Ad Delivery

6.3.1.1 Publication from the Ad Server

The Ad Server SHALL supply the Ads to the DCD Server according to the 'publication-methods' in the channel metadata as described in section 8.2.2.2.4 of [DCD-TSS]. The 'publication-methods' indicates the methods that the Ad Server uses to publish Ad to the DCD Server ("pull": by the DCD Server pulling the updates, "push": by the Ad Server pushing the updates) via DCD-CPDE interface. These publication methods apply to Ad publication from the Ad Server to the DCD Server and they are unrelated to the method of Ad delivery from the DCD Client to the Ad Engine.

If the Ad Server sets the pull method for the publication to the DCD Server in the channel metadata, the DCD Server requests Ads using the DCD-CPDE ContentUpdateRequest message as described in section 7.2.2.2 of [DCD-TSS]. This request can occur according to the 'publication-schedule' in the channel metadata as described in section 8.2.2.2.4 of [DCD-TSS] or following Ad request from the Ad Engine through the DCD Client. The Ad Server SHALL send the DCD-CPDE ContentUpdateResponse message including Ads to the DCD Server.

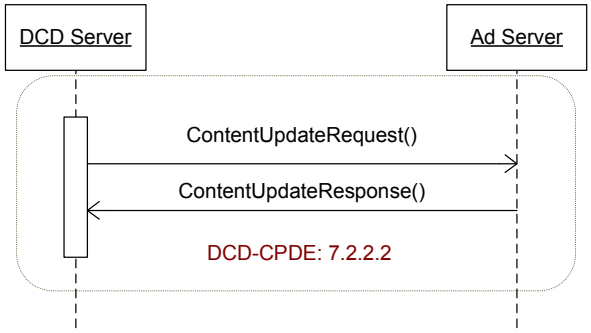

Figure 13Figure 2: MobAd over DCD end to end flow.

If the Ad Server sets the push method for the publication to the DCD Server, the Ad Server SHALL send the DCD-CPDE ContentUpdate message including Ads as described in section 7.2.2.3 of [DCD-TSS].

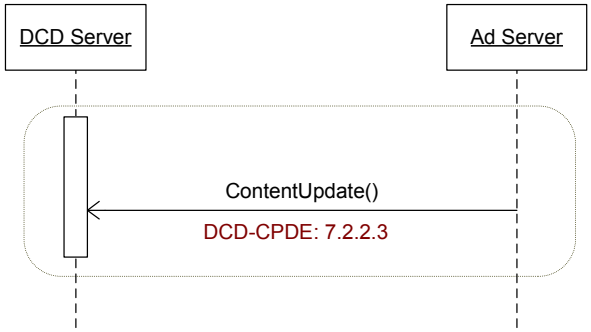

Figure 14Figure 2: MobAd over DCD end to end flow.

6.3.1.2 Push Ad Delivery to the Ad Engine

In case of push Ad delivery, the Ad is pushed according to the 'push-schedule' in the channel metadata as described in section 8.2.2.2.4 of [DCD-TSS]. The 'push-schedule' indicates the set of schedules upon which the DCD Server should push channel updates via push. The Ad Server SHALL send Ads over the DCD-CPDE ContentUpdate message to the DCD Server as described in section 7.2.2.3 of [DCD-TSS] and the DCD Server pushes Ads to the DCD Client, which subscribed to the Ad Channel. When DCD Client receives pushed Ads, the DCD Client SHALL send the DCD-CADE Content message to the Ad Engine as described in section 7.3.2.2 of [DCD-TSS].

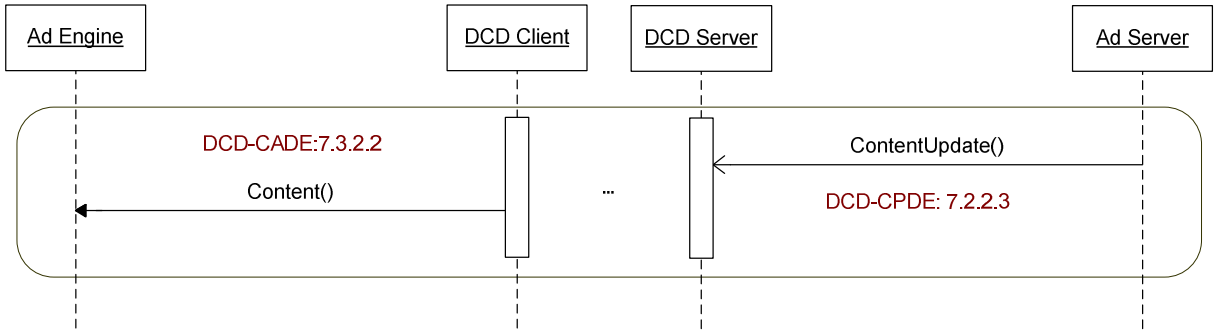

Figure 15Figure 2: MobAd over DCD end to end flow.

6.3.1.3 Associating DCD Content with Advertising

In order to associate DCD Content and Ads using Content Metadata, the DCD content provider or the DCD Server SHALL use Content Metadata parameter *aux-content-link* as defined in section 8.3.2 of [DCD-TSS].

Upon receiving DCD content item with non-empty value of *aux-content-link* metadata parameter, the DCD Client retrieves associated content item (e.g. ads) from the location specified by *aux-content-link* as defined in section 6.1.5.6 of [DCD-TSS]. DCD Client returns first content item to the target DECA and stores the associated Ad until it is requested by the Ad Engine.

If ContentRequest message from DECA (see section 6.3.1.2) contains content-id or content-address attributes, the DCD Client SHALL validate whether the requested content item is already downloaded to the device and provide this content item to DECA. If downloaded content item was associated through *aux-content-link* with the content item obtained by other DECA, the DCD Client MAY validate that requesting DECA is an Ad Engine.

Additionally, there is another valid scenario where advertising is delivered first (e.g. via push bearer) and associated content is linked using *aux-content-link* parameter and pulled by DCD Client upon receiving of advertising.

6.3.2 Suspend and Resume

The behavior and normative requirements described in this section are conditional upon whether the suspend and resume transactions are supported by the DCD Enabler implementations.

6.3.2.1 Suspend and Resume initiated by Ad Server

When the Ad Server chooses to suspend a DCD Channel for Ad Engines, the Ad Server SHALL send the DCD-CPDE ChannelSuspendNotification message to the DCD Server as described in section 7.2.2.1.5 of [DCD-TSS]. Upon receiving notification of channel suspension from the DCD Server, the DCD Client SHALL send the DCD-CADE ChannelSuspendNotification message to the Ad Engine as described in section 7.3.2.4.5 of [DCD-TSS]. Upon this suspend notification, the Ad Engine SHALL abstain from Ad requests to the Ad Server until the latter resumes Ad delivery. To resume Ad delivery, the Ad Server SHALL send the DCD-CPDE ChannelResumeNotifcation message to the DCD Server as described in section 7.2.2.1.7 of [DCD-TSS]. The DCD Server SHALL send the DCD-CPDE ChannelResumeConfirmation message to the Ad Server as described in section 7.3.2.4.7 of [DCD-TSS] and pass the resume notification to the DCD Client. The DCD Client SHALL send the DCD-CADE ChannelResumeNotification message to the Ad Engine. Following this resume notification, the Ad Engine can request Ad delivery again.

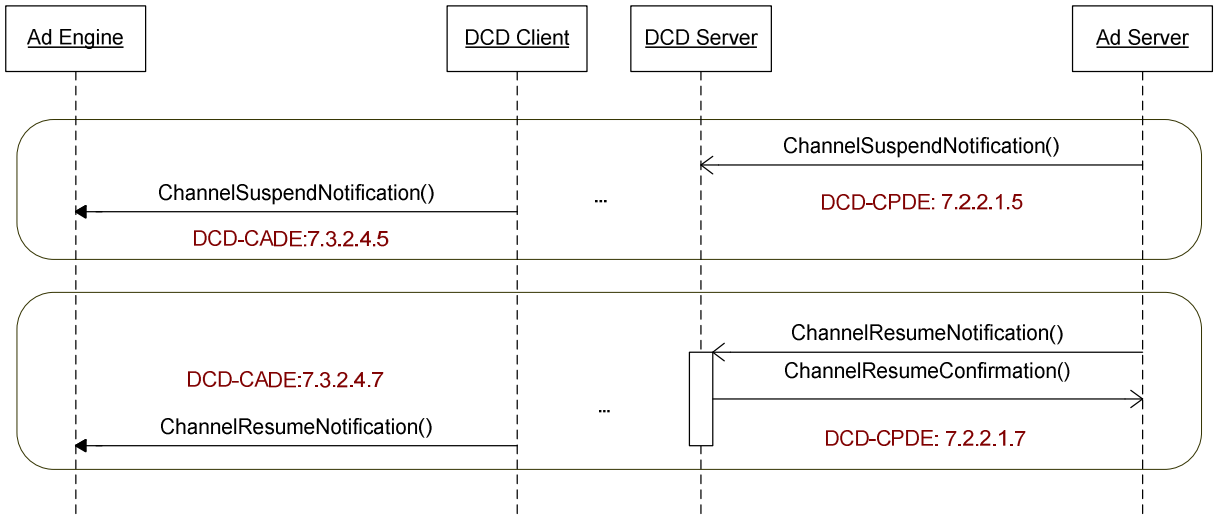

Figure 16Figure 2: MobAd over DCD end to end flow.

6.3.2.2 Suspend and Resume initiated by Ad Engine

When the Ad Engine chooses to suspend a DCD Channel, the Ad Engine SHALL send DCD-CADE ChannelSuspendRequest message as described in section 7.3.2.4.1 of [DCD-TSS]. Following this request, the DCD Server SHALL send the DCD-CPDE ChannelSuspendRequest message to the Ad Server as described in section 7.2.2.1.1 of [DCD-TSS]. Upon this request message, the Ad Server SHALL return the DCD-CPDE ChannelSuspendResponse message to the DCD Server. Finally, the DCD Client SHALL send the DCD-CADE ChannelSuspendResponse message to the Ad Engine. Upon this suspend request, the Ad Engine SHALL abstain from Ad requests to the Ad Server until the Ad Engine resumes the Ad delivery.

When the Ad Engine chooses to resume Ad delivery, it SHALL send DCD-CADE ChannelResumeRequest message as described in section 7.3.2.4.3 of [DCD-TSS]. If the channel delivery from the Ad Server was suspended, the DCD Server SHALL send the DCD-CPDE ChannelResumeRequest message to the Ad Server as described in section 7.2.2.1.3 of [DCD-TSS]. Upon this request message, the Ad Server SHALL return the DCD-CPDE ChannelResumeResponse message to the DCD Server. Finally, the DCD Client SHALL send the DCD-CADE ChannelResumeResponse message to the Ad Engine. Following this message, the Ad Engine can request Ad delivery again.

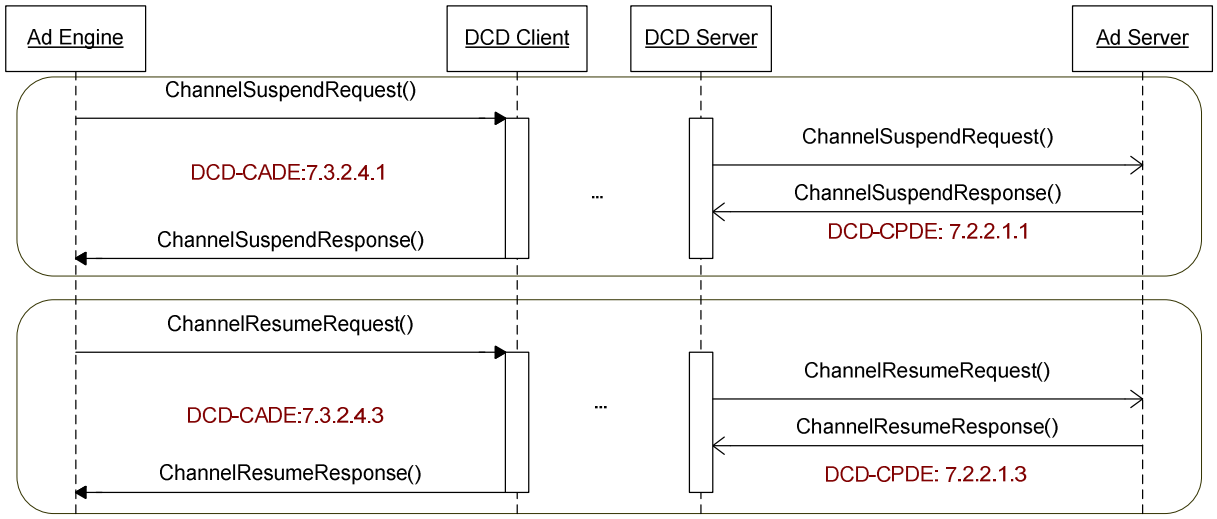

Figure 17Figure 2: MobAd over DCD end to end flow.

6.3.3 MobAd Rule Delivery

MobAd Rule content is opaque to the DCD Enabler, therefore the message flow for rule delivery is based on push delivery transaction.

The Ad Server SHALL send the MobAd Rule content over the DCD-CPDE ContentUpdate message to the DCD Server as described in section 7.2.2.3 of [DCD-TSS] and the DCD Server pushes the MobAd Rule content to the DCD Client. When the MobAd Rule content arrives at the DCD Client, the DCD Client SHALL send the DCD-CADE Content message to the Ad Engine as described in section 7.3.2.2 of [DCD-TSS].

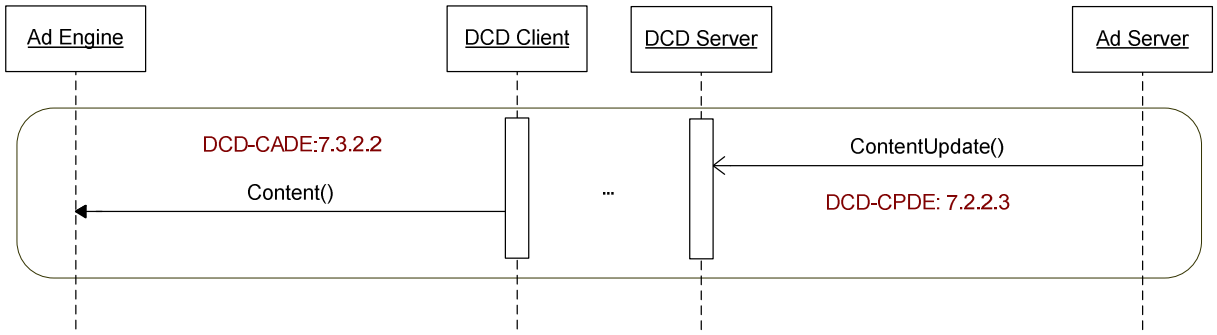

Figure 18Figure 2: MobAd over DCD end to end flow.

7. Application Profile

The Ad Engine provides an application profile to the DCD Enabler as part of the registration process. The application profile (AP) is defined in section 8 of [DCD-TSS].

The goal of the application profile is *"the set of static definitions and rules that allow the DCD enabler to handle the delivery of the DCD Content for a particular DCD-Enabled Client Application"*. In the context of MobAd, the Ad Engine is the only allowed DECA and will provide an application profile to the DCD enabler.

The Ad Engine serves as a mediator between the Ad Apps and the DCD client. The application profile can be generated based on the preferences known to the Ad Engine (user interest, devices capabilities, installed Ad Apps), and by information retrieved from or passed by the Ad Apps (e.g. Ad formats, mime-types, topics, etc).

With regard to the application profile, Ad Engine SHALL support one of the following two options:

- The Ad Engine can register once providing a consolidated application profile with unique DECA application identifier for the Ad Engine. Such application profile reflects combined preferences of all Ad Apps on device and is used to get advertisement of relevance for all Ad Apps. When the Ad Engine receives advertisement content from a subscribed channel, it routes this content to the appropriate Ad App based on the Ad Request and preferences for the Ad App.
- Alternatively, the Ad Engine can register multiple times providing different application profiles. Each application profile has unique DECA application identifier that may embed application identifier for the Ad Engine. Such application profiles reflect preferences for a single Ad App or for a set of Ad Apps with similar preferences. When the Ad Engine receives advertisement content from a subscribed channel, it routes this content to the appropriate Ad App(s) based on the Ad Request and the mapping of relevant DECA application identifier to corresponding Ad App(s).

Note: As the [MobAd-Core-TS] doesn't define any explicit mechanisms to provide Ad App capabilities to the Ad Engine in order to create or obtain an Application Profile, it is assumed that such information could be derived from communications exchanged between these two components, could be computed by the Ad Engine or could be provided by proprietary means (e.g. device runtime inspecting JAR file manifest of the installed Ad App).

8. Interfaces

8.1 MobAd-1 interface

In [MobAd-Core-TS], MobAd-1 is defined as 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.

In this specification the MobAd-1 interface is out of scope and is used and defined between the Ad Engine and the Ad App(s) as per [MobAd-Core-TS] section 5.2.

8.2 MobAd-2 interface

In [MobAd-Core-TS], 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).

In this specification the MobAd-2 interface is out of scope and is used and defined between the Ad Server and the SP App(s) as per [MobAd-Core-TS] section 5.3.

8.3 Delv-1 interface mapping to DCD interfaces

In [MobAd-Core-TS], this interface is used by the Ad Server to push Ad(s), Ad metadata, contextual data, Notifications and MobAd Rules to Ad Engine.

As described in section 5.1 the Delv-1 messages are transmitted from the Ad Server to the Ad Engine via the DCD Server over the DCD-CPDE and DCD-CPR interfaces and via the DCD Client over the DCD-CAR and the DCD-CADE interfaces.

This section describes the mapping of the Delv-1 messages to DCD-CPDE, DCD-CPR messages on the server side and DCD-CAR and DCD-CADE messages on the client side.

8.3.1 AdServerPushAds message

8.3.1.1 Ad Server to DCD Server

The Ad Server provides Ads and Ad Metadata to the DCD server according to the DCD Content Publication method (defined in section 7.2.2.3 or in section 7.2.2.2 of [DCD-TSS]) over the DCD-CPDE interface as described in section 6.3.1.1.

All provided parameters of the AdServerPushAds message defined in section 5.5.1.1of [MobAd-Core-TS] SHALL be included in the Content-Package structure of the ContentUpdate message (defined in section 7.2.2.3.1 of [DCD-TSS]) according to the following:

- The Content-id in the Content-Metadata element of the Content-Package of the ContentUpdate message SHALL be identical to the value of the Ad-ID
- The replace-content-id in the Content-Metadata element of the Content-Package of the ContentUpdate message SHALL take the value of the Outdated-AdID to allow Ad cancellation and replacement. Note: if the corresponding Ad is still available in the storage of the DCD server, the DCD server should delete and replace the cancelled Ad in its storage.

- All other provided parameters SHALL be included in the Content element of the Content-Package of the ContentUpdate message

8.3.1.2 DCD Client to Ad Engine

Upon reception of the ContentUpdatePush message defined in section 7.1.2.1.1 of [DCD-TSS], the DCD Client SHALL send the Content Message defined in section 7.3.2.2.1 of [DCD-TSS] to the Ad Engine.

All provided parameters of the AdServerPushAds message SHALL be included in the Content element of the Content message according to the following:

- The AdID SHALL be used as the value of the Content-id in the Content-Metadata element of the Content message
- The Outdated-AdID SHALL be used as the value of the replace-content-id in the Content-Metadata element of the Content message.
- Note: if the ContentUpdatePush message received by the DCD client contains a replace-content-id element, the DCD Client SHALL delete and replace the cancelled Ad in the DCD client controlled storage, if applicable prior to sending the Content message to the Ad Engine.

8.3.2 PushMobAdRule message

MobAd Rules are delivered via the DCD enabler as application specific content, which is opaque for DCD.

8.3.2.1 Ad Server to DCD Server

The Ad Server provides MobAd Rules to the DCD server according to the DCD Content Publication method defined in section 7.2.2.3 of [DCD-TSS] over the DCD-CPDE interface.

All parameters of the PushMobAdRule message defined in section 5.5.2.1of [MobAd-Core-TS] SHALL be included in the Content structure of the ContentUpdate message defined in section 7.2.2.3.1 of [DCD-TSS]

8.3.2.3 DCD Client to Ad Engine

Upon reception of the ContentUpdatePush message defined in section 7.1.2.1.1 of [DCD-TSS], the DCD Client SHALL send the Content message defined in section 7.3.2.2.1 of [DCD-TSS] to the Ad Engine.

8.3.3 AdServerPushNotif message

The Ad Server push notifications for cancelation of Ads/Campaign correspond to an existing method in DCD for which the binding is defined below.

Other Ad Server push notifications should be delivered similarly to MobAdRule as specified in section 8.3.2.

8.3.3.1 Notification for cancelation of Ads/Campaign

8.3.3.1.1 Ad Server to DCD Server

The Ad Server SHALL provide to the DCD server IDs of Ads that need to be cancelled according to the DCD Content Publication method (defined in section 7.2.2.3 of [DCD-TSS]) over the DCD-CPDE interface.

The provided parameters of the AdServerPushNotif message defined in section 5.5.1.1of [MobAd-Core-TS] SHALL be included in the Content-Package structure of the ContentUpdate message defined in section 7.2.2.3.1 of [DCD-TSS] according to the following:

- The replace-content-id attribute in the Content-Metadata element of the Content-Package SHALL take the value of the ID of an Ad that needs to be cancelled
- All other provided parameters SHALL be included in the Content element of the Content-Package

8.3.3.1.2 DCD Client to Ad Engine

Upon reception of the ContentUpdatePush message defined in section 7.1.2.1.1 of [DCD-TSS], the DCD Client SHALL:

- Delete cancelled Ad in the DCD client controlled storage, if applicable
- Send the Content message (defined in section 7.3.2.2.1 of [DCD-TSS]) to the Ad Engine.

8.4 DCD-CPDE

The DCD-CPDE interface is an interface between the Ad Server and the DCD server used to exchange content as described in section 8.3 and 8.4

Other messages exchanged over this interface as defined in [DCD-TSS] MAY be supported by some implementations but are not required for MobAd over DCD.

8.5 DCD-CPR

The DCD-CPR interface is an interface between the Ad Server and the DCD server used to perform:

- Registration and deregistration operations
- subscription operations
- Channel update operation

as described in section 6.1 and 6.2.

Other messages exchanged over this interface as defined in [DCD-TSS] MAY be supported by some implementations but are not required for MobAd over DCD.

8.6 DCD-CADE

The DCD-CADE interface is an interface between the Ad Engine and the DCD Client used to exchange content as described in section 8.3 and 8.4.

Other messages exchanged over this interface as defined in [DCD-TSS] MAY be supported by some implementations but are not required for MobAd over DCD.

8.7 DCD-CAR

The DCD-CAR interface is an interface between the Ad Engine and the DCD client used to perform:

- Ad Engine registration and deregistration operations
- Channel subscription and un-subscription operations
- Subscription personalisation operation

as described in section 6.1 and 6.2.

Other messages exchanged over this interface as defined in [DCD-TSS] MAY be supported by some implementations but are not required for MobAd over DCD.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-MobAd_DCD_Adaptation-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].

The Mandatory and Optional features of this specification are conditional to the support of MobAd over DCD Adaptation in accordance with the Requirement Column.

B.1 SCR for Ad Engine

Item	Function	Reference	Requirement
MobAd-DCD-E-001-O	Support MobAd over DCD Adaptation		
MobAd-DCD-E-002-M	Support channel registration with DCD Client	Section 6.1.1	MobAd-DCD-E-001-O
MobAd-DCD-E-003-M	Support channel de-registration with DCD Client	Section 6.1.2	MobAd-DCD-E-001-O
MobAd-DCD-E-004-M	Support channel subscription with DCD Client	Section 6.2.2	MobAd-DCD-E-001-O
MobAd-DCD-E-005-M	Support channel unsubscription with DCD Client	Section 6.2.4	MobAd-DCD-E-001-O
MobAd-DCD-E-006-M	Support subscription personalization with DCD Client	Section 6.2.6	MobAd-DCD-E-001-O
MobAd-DCD-E-007-M	Support the reception of Ad delivery from DCD Client	Section 6.3.1.2	MobAd-DCD-E-001-O
MobAd-DCD-E-008-M	Support channel suspend and resume	Section 6.3.2.2	MobAd-DCD-E-001-O
MobAd-DCD-E-009-M	Support the Ad Metrics data reporting through the DCD Enabler	Section 6.3.3	MobAd-DCD-E-001-O
MobAd-DCD-E-010-O	Support the reception of MobAd Rules through the DCD Enabler	Section 6.3.3	MobAd-DCD-E-001-O

B.2 SCR for Ad Server

Item	Function	Reference	Requirement
MobAd-DCD-S-001-O	Support MobAd over DCD Adaptation		
MobAd-DCD-S-002-M	Support channel registration with DCD Server	Section 6.1.3	MobAd-DCD-S-001-O
MobAd-DCD-S-003-M	Support channel deregistration with DCD Server	Section 6.1.4	MobAd-DCD-S-001-O
MobAd-DCD-S-004-O	Support Ad Server initiated Ad Engine channel subscription	Section 6.2.1	MobAd-DCD-S-001-O
MobAd-DCD-S-005-O	Support Ad Server initiated Ad Engine channel unsubscription	Section 6.2.3	MobAd-DCD-S-001-O
MobAd-DCD-S-006-O	Support channel update with DCD Server	Section 6.2.5	MobAd-DCD-S-001-O
MobAd-DCD-S-007-M	Support Ad publication to DCD Server	Section 6.3.1.1	MobAd-DCD-S-001-O
MobAd-DCD-S-008-O	Support channel suspend and resume	Section 6.3.2.1	MobAd-DCD-S-001-O
MobAd-DCD-S-009-O	Support the delivery of MobAd Rules through DCD Enabler	Section 6.3.3	MobAd-DCD-S-001-O
MobAd-DCD-S-010-M	Support the reception of the Ad Metrics data through DCD Enabler	Section 6.3.3	MobAd-DCD-S-001-O

Appendix C. MobAd-3 interface mapping to DCD interfaces (Informative)

In [MobAd-Core-TS], MobAd-3 is defined as 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).

In the context of MobAd over DCD, the Ad Engine's AdRequests to the Ad Server translates into ContentSubmitRequest over DCD-CADE interface, as specified in section 7.3.2.3 of [DCD-TSS]. This request results in the subsequent request over DCD-1 interface (from the DCD Client to the DCD Server) and may further result in the ContentSubmitRequest over DCD-CPDE interface, as specify in section 7.2.2.2 of [DCD-TSS] to the Ad Server.

Delivery of the response may be provided over the DCD-1 or DCD-2 interface depending on the Service Provider policies.

This section describes how the MobAd-3 messages are transmitted from the Ad Engine to the DCD Client over the DCD-CADE interfaces and from the DCD Server to the Ad Server to over the DCD-CPDE interface.

C.1 Pull delivery to the Ad Engine

In case of the pull Ad delivery, the channel metadata should have the 'on-demand-pull-allowed' element set as "TRUE" to allow the Ad Engine to request the Ads from the DCD Client. This request can be triggered by an Ad App or initiated by the Ad Engine (e.g. for Ad pre-fetching).

There are two possible methods for Ad Engine to obtain available Ads from an Ad channel:

1. The Ad Engine SHALL send the DCD-CADE ContentRequest message to the DCD Client as described in section 7.3.2.2 of [DCD-TSS]. As the DCD-CADE ContentRequest message doesn't allow providing information for Ad selection, this method is used when subscribed Ad Channel is specialized according to provided mime-types and/or content-types in the channel metadata (i.e. all Ads available in this channel are applicable) or when the Ad preferences were already provided to the Content Provider as a subscription filter. In the latter case, the DCD Server SHALL only return Ads associated with the subscription-id matching the preferences provided at channel subscription.

2. The Ad Engine SHALL send the DCD-CADE ContentSubmitRequest message including the element 'Submit-Package' for the Ad request (e.g. 'ContextualData' in [MobAd-Core-TS]) as described in section 7.3.2.3 of [DCD-TSS]. The content of 'Submit-Package' element is targeted for the Ad Server and opaque for the DCD Enabler. This method is better suited for dynamic Ad preferences (e.g. contextual data associated with the displayed content or state of the device).

Following the content request from the DCD Client, and if requested data is not available at the DCD Server (e.g. published by the Ad Server), the DCD Server SHALL send the DCD-CPDE ContentUpdateRequest message to the Ad Server as described in section 7.2.2.2 of [DCD-TSS]. Consecutively, the Ad Server SHALL return the DCD-CPDE ContentUpdateResponse message to the DCD Server.

Finally, the DCD Client SHALL send to Ad Engine the DCD-CADE Content message including Ad(s) in the 'Content' element as described in section 7.3.2.2 of [DCD-TSS].

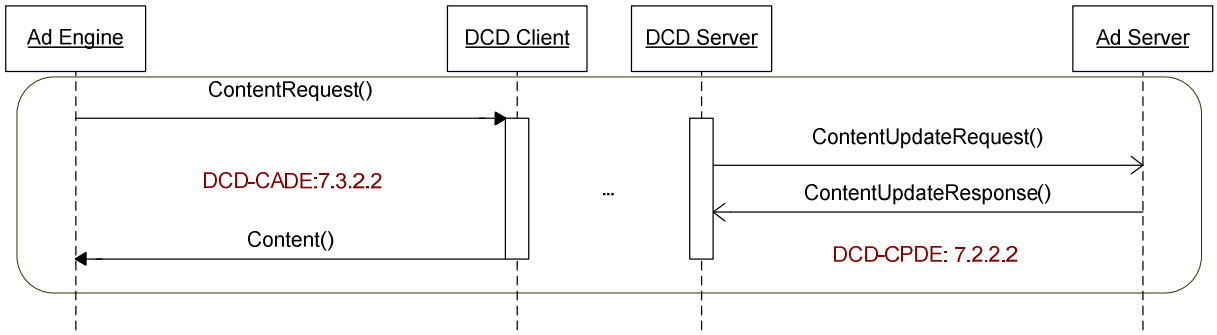

Figure 19Figure 2: MobAd over DCD end to end flow.

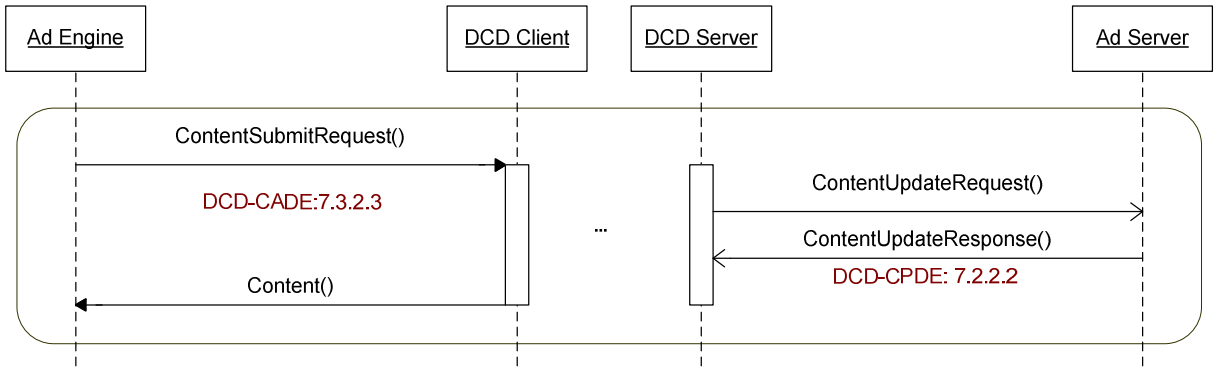

Figure 20Figure 2: MobAd over DCD end to end flow.

C 1.1 Ad Engine to DCD Client

The Ad Engine SHALL provide the AdEngineAdRequest message parameters to the DCD Client as a ContentSubmitRequest or ContentRequest message (specified in section 7.3.2.3 and 7.3.2.2of [DCD-TSS]) over DCD-CADE interface as described in section 6.3.1.2.

The provided parameters of the AdEngineAdRequest message defined in section 5.4.1.1of [MobAd-Core-TS] SHALL be included in the ContentSubmitRequest or ContentRequest message according to the following mapping:

- The DCD Application-ID SHALL be the ID provided by Ad Engine at registration (i.e. Ad-Engine ID or ID representing the subset of Ad Apps relevant to this registration)
- All other provided parameters of the AdEngineAdRequest message SHALL be included in the Submit-Package structure of the ContentSubmitRequest or ContentRequest.

C 1.2 DCD Server to Ad Server

Upon reception of the ContentSubmitRequest the DCD Server SHALL send to the Ad Server a ContentUpdateRequest as specified in section 7.2.2.2 of [DCD-TSS] over the DCD-CDPE interface.

C.2 AdEngineAdResponse Message

C 2.1 Ad Server to DCD Server

In response to the Ad Request, the Ad Server SHALL provide the DCD Server with the AdEngineAdResponse Message parameters as a ContentUpdateResponse message over the DCD-CPDE interface, as specified in section 7.2.2.2 of [DCD-TSS]

All provided parameters of the AdEngineAdresponseMessage SHALL be included in the ContentPackage of the ContentUpdateResponse with the following limitation:

- The AdID SHALL be used as the value of the Content-id in the content-metadata element of the ContentPackage
- The replace-content-id in the Content-Metadata element of the Content-Package of the ContentUpdateResponse message SHALL take the value of the Outdated-AdID to allow Ad cancellation and replacement. Note: if the corresponding Ad is still available in the storage of the DCD server, the DCD server should delete and replace the cancelled Ad in its storage.
- All other provided parameters SHALL be included in the Content element of the Content-Package

C 2.2 DCD client to Ad Engine

Upon reception of the ContentUpdate Response, the DCD client SHALL send to the Ad Engine a Content message as specified in section 7.3.2.2 of [DCD-TSS] over the DCD-CAD interface, which corresponds to the AdEngineAdResponse message.

All provided parameters of the Content message SHALL be included in the AdEngineAdResponse message with the following limitation:

- The AdID SHALL be used as the value of the Content-id in the Content-Metadata element
- All other provided parameters SHALL be included in the Content element

C.3 AdEngineMetricReport Message

C 3.1 Metric Reporting Flow

Metrics reporting content is opaque to the DCD Enabler, therefore the message flow for Metrics reporting utilizes DCD Content Submission transaction.

The Ad Engine SHALL send the DCD-CADE ContentSubmitRequest message including the element ‘Submit-Package’ for the Metrics Report as described in section 7.3.2.3 of [DCD-TSS].

Following the Metrics Report submission from the DCD Client, the DCD Server SHALL send the DCD-CPDE ContentUpdateRequest message to the Ad Server as described in section 7.2.2.2 of [DCD-TSS]. Consecutively, the Ad Server SHALL return the DCD-CPDE ContentUpdateResponse message to the DCD Server.

Finally, the DCD Client SHALL send the DCD-CADE ContentSubmitConfirmation message to the Ad Engine as described in section 7.3.2.3 of [DCD-TSS].

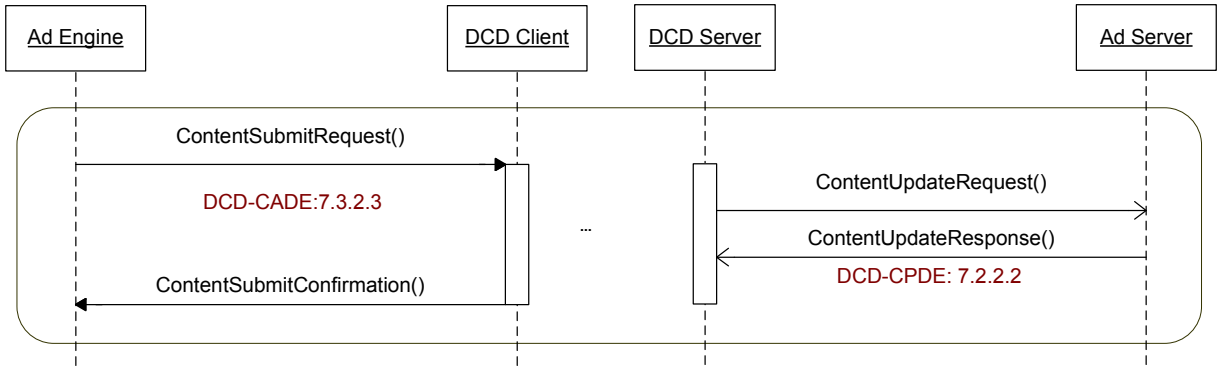

Figure 21Figure 2: MobAd over DCD end to end flow.

C 3.2 AdEngine to DCD client

The Ad Engine SHALL provide the AdEngineMetricReport message to the DCD Client as a ContentSubmitRequest message specified in section 7.3.2.3 of [DCD-TSS] over DCD-CADE interface.

The provided parameters of the AdEngineMetricReport message defined in section 5.4.2.1 of [MobAd-Core-TS] SHALL be included in the ContentSubmitRequest message according to the following mapping:

- The DCD Application-ID SHALL take the ID provided by Ad Engine at registration (i.e. Ad-Engine ID or ID representing the subset of Ad Apps relevant to this registration)
- All other provided parameters of the AdEngineMetricReport message SHALL be included in the Submit-Package structure of the ContentSubmitRequest.

C 3.3 DCD Server to Ad Server

Upon reception of the ContentSubmitRequest the DCD Server SHALL send to the Ad Server a ContentUpdateRequest as specified in section 7.2.2.2 of [DCD-TSS] over the DCD-CDPE interface.

C.4 AdEngineRuleRequest Message

The AdEngineRuleRequest message SHALL be provided as a ContentSubmitRequest message and the mapping of the parameters apply as defined in section 8.4.1.

C.5 AdEngineRuleResponse Message

The AdEngineRuleRequest message SHALL be provided as a ContentUpdateResponse message and the mapping of the message parameters apply as defined in section 8.4.2