



Dynamic Content Delivery Architecture

Candidate Version 1.0 – 23 Dec 2008

Open Mobile Alliance
OMA-AD-DCD-V1_0-20081223-C

Use of this document is subject to all of the terms and conditions of the Use Agreement located at <http://www.openmobilealliance.org/UseAgreement.html>.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at <http://www.openmobilealliance.org/ipr.html>. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2008 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

Contents

1. SCOPE (INFORMATIVE)	5
2. REFERENCES	6
2.1 NORMATIVE REFERENCES	6
2.2 INFORMATIVE REFERENCES	7
3. TERMINOLOGY AND CONVENTIONS	8
3.1 CONVENTIONS	8
3.2 DEFINITIONS	8
3.3 ABBREVIATIONS	10
4. INTRODUCTION (INFORMATIVE)	11
4.1 PLANNED PHASES	13
4.2 SECURITY CONSIDERATIONS	13
4.2.1 Authentication and Authorization	13
4.2.2 Delivery Security	13
4.2.3 Content Protection	13
4.2.4 Content and Subscription Information Privacy	14
4.3 ACTORS	14
4.3.1 DCD-Enabled Client Application	14
4.3.2 DCD Content Provider	15
5. ARCHITECTURAL MODEL	16
5.1 DEPENDENCIES	16
5.2 ARCHITECTURAL DIAGRAM	16
5.3 DCD CONTENT PACKAGING	18
5.4 REGISTRATION WITH THE DCD ENABLER	20
5.5 DCD PROCESSES, FUNCTIONAL COMPONENTS AND INTERFACES	21
5.5.1 DCD Processes	21
5.5.2 DCD Functional Entities	24
5.5.3 External Enabler Dependencies	29
5.5.4 Interfaces	30
5.6 FLOWS	32
5.6.1 Registration of the DCD Enabled Client Application	32
5.6.2 Registration of the DCD Enabled Client Application without interaction with the DCD Server	33
5.6.3 Channel Subscription	34
5.6.4 Notification on External Channel Subscription	35
5.6.5 DCD Content Delivery	36
5.6.6 Deregistration of the DCD Enabled Client Application	40
5.6.7 Lifecycle of the DCD Enabled Client Application	41
6. DCD ENABLER PROTOCOL STACK	44
6.1 DCD ENABLER PROTOCOL STACK	44
APPENDIX A. OTHER OMA ENABLER ENTITY USE BY DCD (INFORMATIVE)	46
A.1 OMA DEVICE MANAGEMENT ENABLER	46
A.1.1 Device Provisioning and Management Client	46
A.1.2 Device Provisioning and Management Server	46
A.2 OMA CHARGING ENABLER	47
A.2.1 Charging Entity	47
A.3 OMA PRESENCE ENABLER	47
A.3.1 Presence Server	47
A.4 OMA LOCATION ENABLER	47
A.4.1 Location Server	47
A.5 OMA CATEGORIZATION-BASED CONTENT SCREENING ENABLER	48
A.5.1 Content Screening Entity	48
A.6 OMA CLIENT-SIDE CONTENT SCREENING FRAMEWORK ENABLER	48

A.6.1 Scan Engine 48

A.7 OMA USER AGENT PROFILE ENabler..... 48

A.7.1 User Agent Profile Storage Server 48

A.8 OMA DRM ENabler..... 49

APPENDIX B. DCD OVER BCAST ARCHITECTURE..... 50

B.1 ARCHITECTURAL DIAGRAM 50

B.2 BCAST REFERENCE POINTS..... 50

B.2.1 BCAST-5 Reference Point 51

B.2.2 BCAST-6 Reference Point 51

B.2.3 BCAST-7 Reference Point 51

B.3 DCD OVER BCAST INTERFACES 52

B.3.1 DCD-BCAST-1 Interface 52

B.3.2 DCD-BCAST-2 Interface 52

B.3.3 DCD-BCAST-3 Interface 52

B.4 DCD OVER BCAST METADATA..... 52

B.4.1 Inclusion of BCAST Access Information in DCD Metadata 52

B.4.2 Supported BCAST Access Information Formats 53

APPENDIX C. CHANGE HISTORY (INFORMATIVE)..... 54

C.1 APPROVED VERSION HISTORY 54

C.2 DRAFT/CANDIDATE VERSION 1.0 HISTORY 54

Figures

Figure 1: DCD Enabler Interactions with External Enablers and Applications..... 12

Figure 2: DCD Enabler Architecture..... 17

Figure 3: DCD Packaging – Envelope Model..... 19

Figure 4: Registration with the DCD Enabler..... 20

Figure 5: Registration of DCD Enabled Client Application 32

Figure 6: Registration of the DCD Enabled Client Application without DCD Client interaction with the DCD Server..... 34

Figure 7: Channel Subscription by a DCD-Enabled Client Application 35

Figure 8: Notification of External Subscription for a DCD Channel..... 35

Figure 9: Pull-based Content Delivery over DCD-1 Interface..... 36

Figure 10: Push-based Content Delivery over DCD-2 Interface 37

Figure 11: Pull based Content Delivery over DCD-1 Interface with Content Availability Notification over DCD-2 Interface..... 38

Figure 12: Suspension and Resumption of Content Delivery Channels..... 39

Figure 13: Deregistration of DCD Enabled Client Application 40

Figure 14: Lifecycle of the DCD Enabled Client Application using P2P content delivery 42

Figure 15: DCD Enabler Protocol Stack..... 44

Figure 16: DCD over BCAST Functional Architecture Diagram 50

Tables

Error! No table of figures entries found.

1. Scope

(Informative)

This document defines the architecture of Dynamic Content Delivery (DCD) Enabler. This architecture is based on the requirements listed in the DCD Requirement Document [OMA-DCD-RD].

2. References

2.1 Normative References

- [OMA-BCAST-AD] “Mobile Broadcast Services Architecture”, Open Mobile Alliance™, OMA-AD-BCAST-V1_0, URL:<http://www.openmobilealliance.org/>
- [OMA-Browsing] “OMA Browsing v2.3”, Open Mobile Alliance™, OMA-ERP-Browsing-V2_3, URL:<http://www.openmobilealliance.org/>
- [OMA-CBCS-AD] “Categorization-based Content Screening Framework Architecture”, Open Mobile Alliance™, OMA-AD-CBCS-V1_0, URL:<http://www.openmobilealliance.org/>
- [OMA-CP-ARCH] “OMA Client Provisioning v1.1 – Provisioning Architecture Overview”, Open Mobile Alliance™, OMA-WAP-ProvArch-V1_1, URL:<http://www.openmobilealliance.org/>
- [OMA-CSCF-AD] “Client Side Content Screening Framework Architecture”, Open Mobile Alliance™, OMA-AD_Client_Side_CS_FW-V1_0, URL:<http://www.openmobilealliance.org/>
- [OMA-CSCF-TS] “Client Side Content Screening Framework Specification”, Open Mobile Alliance™, OMA-TS-Client_Side_CS_FW-V1_0, URL:<http://www.openmobilealliance.org/>
- [OMA-DCD-RD] “Dynamic Content Delivery Requirements”, Open Mobile Alliance™, OMA-RD-DCD-V1_0, URL:<http://www.openmobilealliance.org/>
- [OMA-DCD-TS-Semantics] “Dynamic Content Delivery Technical Specification – Session and Transactions”, Open Mobile Alliance™, OMA-TS-DCD_Semantics-V1_0, URL:<http://www.openmobilealliance.org/>
- [OMA-DM-AD] “Device Management Architecture v1.0”, Open Mobile Alliance™, OMA-AD-DM-V1_0, URL:<http://www.openmobilealliance.org/>
- [OMA-DPE-RD] “Device Profile Requirements Version 1.0”, Open Mobile Alliance™, OMA-RD-DPE-V1_0, URL:<http://www.openmobilealliance.org/>
- [OMA-DRM-AD] “Digital Rights Management (DRM) Architecture v2.0”, Open Mobile Alliance™, OMA-AD-DRM-V2_0, URL:<http://www.openmobilealliance.org/>
- [OMA-FUMO-AD] “Firmware Update Management Object Architecture v1.0”, OMA-AD-FUMO-V1_0, Open Mobile Alliance™, URL:<http://www.openmobilealliance.org/>
- [OMA-MCC-AD] “Charging Architecture”, Open Mobile Alliance™, OMA-AD-Charging-V1_0, URL:<http://www.openmobilealliance.org/>
- [OMA-MLP3.2] “Mobile Location Protocol v3.2”, Open Mobile Alliance™, OMA-TS-MLP-V3_2, URL:<http://www.openmobilealliance.org/>
- [OMA-MLS-AD] “Mobile Location Service Architecture v1.0”, OMA-AD-MLS-V1_0, Open Mobile Alliance™, URL:<http://www.openmobilealliance.org/>
- [OMA-OSE] “OMA Service Environment”, Open Mobile Alliance™, OMA-Service-Environment-V1_0_2, URL: <http://www.openmobilealliance.org/>
- [OMA-PRS-IMPS-AD] “Presence IMPS Architecture v1.3”, OMA-AD-IMPS-V1_3, Open Mobile Alliance™, URL:<http://www.openmobilealliance.org/>
- [OMA-PRS-SIMPLE-AD] “Presence SIMPLE Architecture v1.0”, OMA-AD-Presence_SIMPLE-V1_0, Open Mobile Alliance™, URL:<http://www.openmobilealliance.org/>
- [OMA-PRS-SIMPLE] “Presence SIMPLE Specification”, Open Mobile Alliance™, OMA-TS-Presence_SIMPLE-V1_0, URL:<http://www.openmobilealliance.org/>
- [OMA-PUSH-AD] “Push Architecture”, Open Mobile Alliance™, OMA-AD-Push-V2_2, URL:<http://www.openmobilealliance.org/>

[OMA-PUSH-OTA]	“Push Over The Air”, Open Mobile Alliance™, OMA-WAP-TS-PushOTA-V2_1, URL: http://www.openmobilealliance.org/
[OMA-PUSH-PPG]	“Push Proxy Gateway Service”, Open Mobile Alliance™, OMA-WAP-TS-PPGService-V2_1, URL: http://www.openmobilealliance.org/
[OMA-SIP_PUSH]	“Push using SIP”, Open Mobile Alliance™, OMA-TS-SIP_Push, URL: http://www.openmobilealliance.org/
[OMA-SIP_PUSH-AD]	“SIP Push Architecture”, Open Mobile Alliance™, OMA-AD-SIP_Push-V1_0, URL: http://www.openmobilealliance.org/
[OMA-SCOMO-AD]	“Software Component Management Object Architecture v1.0”, OMA-AD-SCOMO-V1_0, Open Mobile Alliance™, URL: http://www.openmobilealliance.org/
[OMA-UAPROF]	“OMA User Agent Profile V2.0”, Open Mobile Alliance™, OMA-TS-UAProf-V2_0, URL: http://www.openmobilealliance.org/
[RFC2119]	“Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, URL: http://www.ietf.org/rfc/rfc2119.txt

2.2 Informative References

[3GPP-SMS]	“Technical Realization of the Short Message Service (SMS)”, 3GPP TS23.040, url:http://www.3gpp.org/
[3GPP-CBS]	“Technical Realization of Cell Broadcast Service (CBS)”, 3GPP TS23.041, url:http://www.3gpp.org/
[OMA-DICT]	“OMA Dictionary”, Open Mobile Alliance™, OMA-Dictionary-V2_4, 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

Application	See [OMA-DICT]
Application Profile (AP)	See [OMA-DCD-RD].
Application Registration	See [OMA-DCD-RD].
ATOM	A set of standards for web feeds, including The Atom Syndication Format (an XML language used for web feeds) and the Atom Publishing Protocol (a simple HTTP-based protocol for creating and updating web resources).
Authentication	See [OMA-DCD-RD].
Channel	See [OMA-DCD-RD].
Channel Guide	See [OMA-DCD-RD].
Channel Metadata	The Channel Metadata is a set of static settings and rules for handling delivery of the DCD Content for a particular channel (e.g. delivery, storage, notification rules). Channel Metadata is associated with the channel’s content types.
Channel Registration	The process of establishing the availability of a new DCD Channel.
Channel Resumption	The process of resuming DCD Enabled services for a specific DCD Channel.
Channel Subscription	The process of establishing a Subscription to a DCD Enabled service, represented as a DCD Channel, for a DCD Enabled Client Application.
Channel Suspension	The process of suspending DCD Enabled services for a specific DCD Channel.
Channel Unsubscription	The process of terminating a subscription to a DCD Channel.
Client Activation	The process of establishing or re-establishing a DCD Session.
Client Deactivation	The process of terminating a DCD Session.
Content Metadata	The Content Metadata is a set of dynamic settings and rules for handling delivery of the DCD Content. Content Metadata is associated with a particular unit of content and may include rules and settings for content expiry, replacement, fragmentation, etc.
Content Publication	The process whereby DCD Content Providers make content available for ingestion and subsequent delivery by DCD Servers.
DCD Client	See [OMA-DCD-RD].
DCD Client Device	See [OMA-DCD-RD].
DCD Content	See [OMA-DCD-RD].
DCD Content Provider	See [OMA-DCD-RD].
DCD Distribution and Adaptation	The DCD Distribution and Adaptation is a set of functions that distribute DCD Content and DCD Content Notifications to the DCD Clients, and adapts the data to suit the device and support distribution with bandwidth efficiency.
DCD-Enabled Client Application	See [OMA-DCD-RD].

DCD Server	See [OMA-DCD-RD].
DCD Service	See [OMA-DCD-RD].
DCD Service Provider	See [OMA-DCD-RD].
DCD Subscription and Administration	The DCD Subscription and Administration is a set of functions for administration of the DCD Enabler, including the subscription to the DCD channels. Business aspects of subscription is out of scope of the DCD enabler.
DCD XML	The XML document type used to express DCD messages.
DCD-1	A uni-directional point-to-point interface between the DCD Server and the DCD Client, used by the DCD Client to send content requests and submissions to the DCD Server and to receive DCD Content in response.
DCD-2	A uni-directional interface between the DCD Server and the DCD Client, used by the DCD Server to push notifications and / or content to the DCD Client.
DCD-3	A bi-directional interface between the DCD Server and the DCD Client, is used by the DCD Server and the DCD Client to exchange DCD Subscription and Administration information.
DCD-CADE	A bi-directional interface between the DCD Enabled Client Application and the DCD Client, used by the DCD Client to send notifications and / or content to the DCD Enabled Client Application, and enabling the DCD Enabled Client Application to retrieve content from the DCD Client.
DCD-CAR	A bi-directional interface between the DCD Enabled Client Application and the DCD Client, used by the DCD Enabled Client Application to register with the DCD Client when the application is installed on a terminal.
DCD-CPDE	A bi-directional interface between the DCD Content Provider and the DCD Server, used by the Content Provider to publish content at the DCD Server and by the DCD Server to retrieve content from the Content Provider.
DCD-CPR	A bi-directional interface between the DCD Content Provider and the DCD Server, used by the Content Provider to register new content channels with the DCD Server and to notify the DCD Server about subscription events and vice versa.
Dynamic Content Delivery	See [OMA-DCD-RD].
Enabler	See [OMA-DICT]
Interface	See [OMA-DICT].
Personalization	See [OMA-DCD-RD].
Pull	A service delivery method in which the client initiates a service transaction.
Push	A service delivery method in which the server initiates a service transaction.
Push-OTA	The over-the-air interface via which OMA Push messages are delivered to terminals.
Receive-Only Client Case	Broadcast-related deployment model for DCD in which a DCD Client has no need or ability to transmit requests or responses to DCD transactions.
RSS	See [OMA-DCD-RD].
Subscriber	See [OMA-DICT]
Subscription	See [OMA-DCD-RD].
Subscription Personalization	The process of adapting DCD Enabled services to a particular user's preferences and context.
Subscription Validation	The process of verifying DCD Enabled Client Application agreement with a DCD Server-initiated Channel Subscription action.
Terminal	The mobile device with which an End-User receives and consumes a Broadcast Service.

User	See [OMA-DICT]
User Agent Profile	A document format defined by the OMA UAProf enabler [Error! Reference source not found.], via which a device vendor discloses characteristics of a device and the OMA enablers that it supports.

3.3 Abbreviations

AP	Application Profile
BCAST	The OMA BCAST enabler
CBS	Cell Broadcast Service
CDF	Compound Document Formats
DCD	Dynamic Content Delivery
HTTP	Hypertext Transport Protocol
ID	Identifier
IP	Internet Protocol
MCC	Mobile Charging and Commerce
MIME	Multipurpose Internet Mail Extensions
OMA	Open Mobile Alliance
OMNA	Open Mobile Naming Authority, the operational naming authority established by WAP (as WAP Interim Naming Authority - WINA) and inherited by OMA.
PAP	Push Access Protocol
RFC	Request for Comments
RSS	Really Simple Syndication
UAProf	User Agent Profile
UI	User Interface
URI	Universal Resource Identifier
URL	Universal Resource Locator
WAP	Wireless Application Protocol

4. Introduction

(Informative)

The Dynamic Content Delivery (DCD) Enabler is expected to enhance a mobile user's experience through the periodic delivery of personalised or customized content either on a one-to-one (point-to-point) or one-to-many (broadcast) basis. The delivery of DCD Content may be based on the subscription and preferences of a user, operator or service provider. As a complementary delivery mechanism to the existing mechanisms, e.g. browsing, messaging, etc., it will reuse as much existing technology as possible, while providing the added benefits of delivery control management, and an enhanced user experience. The content delivery will support various network technologies (i.e. network types and/or bearers), and may operate autonomously in the background.

The content types and formats used by DCD will, wherever possible, be consistent with those of the established delivery mechanisms, e.g. XML, RSS, ATOM etc. The DCD Enabler enables an application to be enhanced by making its content available through automatic asynchronous delivery, though it does not specify detailed DCD applications, or how to render the content of those applications. The DCD Enabler is agnostic to different content types used by different applications.

This document defines the architecture of the Dynamic Content Delivery (DCD) Enabler based on the DCD requirements defined in [OMA-DCD-RD].

As part of the DCD Enabler several functions are expected to be standardized. These include, but not limited, to:

- A generic Client framework that allows automated registration of DCD enabled applications;
- The ability to allow customization of the delivery of content by the DCD client in the broadcast scenario;
- The asynchronous delivery of content utilizing both point-to-point and broadcast bearers;
- Content delivery and subscription based notification mechanisms between DCD Client and DCD Server for both point-to-point and broadcast bearers;
- DCD envelope mechanism that allows DCD application and content interoperability between DCD Client and DCD Server.

The DCD Enabler is a part of the OMA Service Environment [OMA-OSE] and designed to interface other OMA enablers. Figure 1 illustrates DCD Enabler interactions with other enablers and applications.

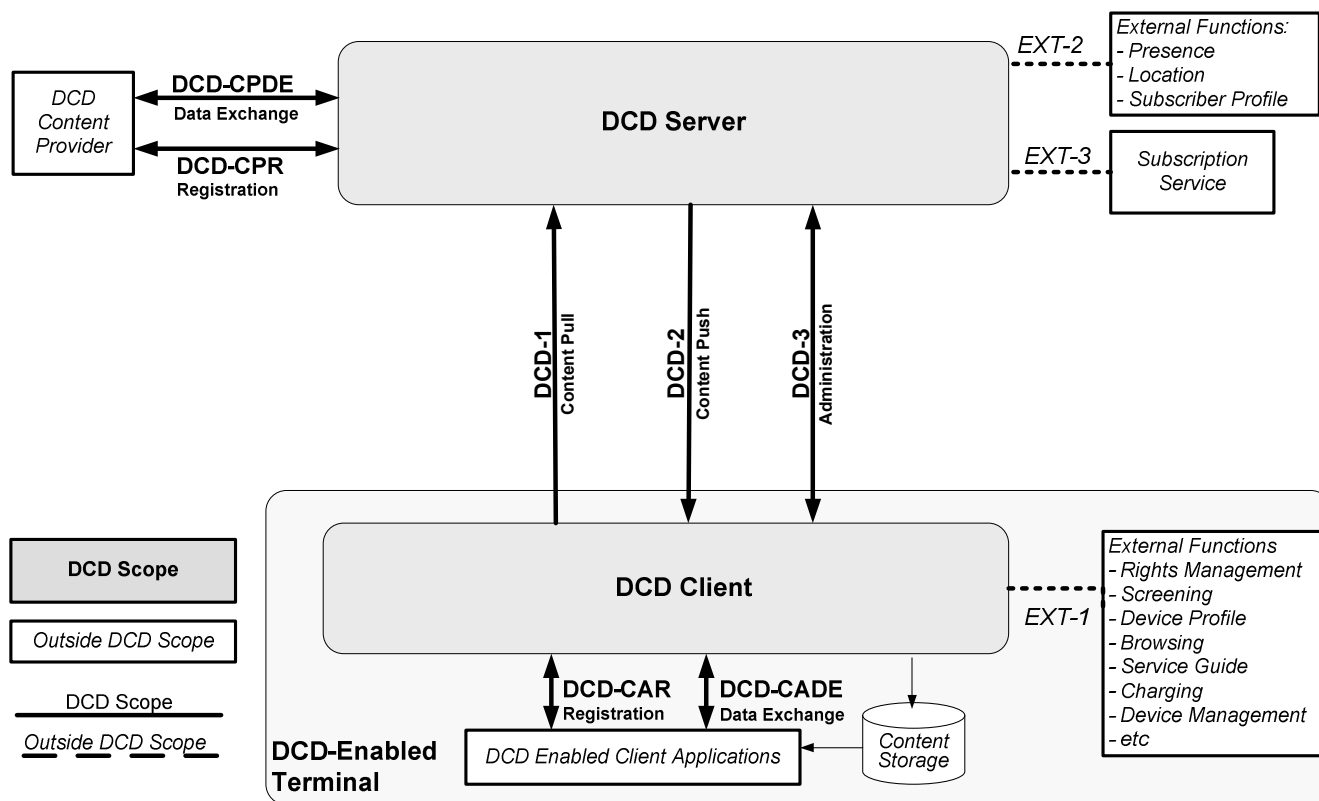


Figure 1: DCD Enabler Interactions with External Enablers and Applications

The structure of this architecture document is as follows.

Section 5 defines the architecture of the DCD Enabler. Section 5.2 introduces the overall DCD architecture diagram and its logical entities and interfaces.

Section 5.3 defines the DCD Content packaging model that makes possible for the DCD Enabler to be content type agnostic.

Section 5.4 presents diagrams for dynamic registration of DCD-Enabled Client Applications and Content Providers’ channels with the DCD Enabler. The registration model supports dynamic plug-in of new applications and channels with the DCD Enabler without interruption of services provided by the enabler.

Section 5.5 describes in further detail the DCD functional entities and their interfaces.

Section 5.6 presents several fundamental signalling flows that show how the basic interactions between the DCD Enabler functions.

In section 6, the DCD Enabler protocol stack is introduced to better describe the end-to-end content delivery flows and how the DCD Enabler handles the content delivery between the DCD Client, DCD Server and the DCD-Enabled Client Application.

The external interfaces EXT-1, EXT-2 and EXT-3 are not defined by the DCD Enabler, but are used by DCD Enabler entities. EXT-2 represents interfaces the DCD Server has to external entities providing information used in fulfilling specific DCD enabler requirements. EXT-3 represents interfaces the DCD Server has to external subscription management systems, for maintaining the current status of DCD channel subscriptions for a user. EXT-1 represents interfaces the DCD Client has to external entities providing information used in fulfilling specific DCD Enabler requirements. Where the interfaces are to

OMA Enabler entities, the use of these interfaces by DCD Enabler entities is described in section 5.5.2, and the OMA-specified interfaces are described.

4.1 Planned Phases

This entire architecture will be implemented in phase 1.

4.2 Security Considerations

The following sections describe the security functions required to be supported by the DCD Enabler.

4.2.1 Authentication and Authorization

4.2.1.1 Secure Session Establishment

The DCD Enabler provides support for a network-independent authentication mechanism between the DCD Client and DCD Server (e.g., Digest Authentication). When required (e.g. for a secure point-to-point delivery scenario), the DCD Client and the DCD Server are mutually authorized to establish a secure session prior to any DCD content delivery requests.

4.2.1.2 DCD-Enabled Client Application Registration

The DCD Enabler supports an authentication and authorization mechanism that allows new DCD-Enabled Client Applications to be registered with a DCD Client, as described in section 5.5.1.2, and the authentication and authorization of the registered DCD-Enabled Client Application with the DCD Server.

4.2.1.3 Two-Way Content and Subscription Security

The DCD Enabler supports the ability to authenticate and authorize a user prior to the delivery from a DCD Server to DCD Client and / or consumption of DCD Content.

The DCD Enabler also supports the ability to ensure that DCD Clients receive DCD content only from authenticated and authorized DCD Servers.

4.2.1.4 Service Configuration

The DCD Enabler ensures that only authorized actors are permitted to update DCD service configuration information (e.g. subscriptions, channel metadata, etc.). When such information resides on a DCD Client Device, a user's confirmation may be requested.

4.2.2 Delivery Security

The DCD Enabler applies measures of security for DCD Content and DCD Content notification delivery, e.g. as required for the used delivery mechanisms (see [OMA-BCAST-AD]).

When required, the DCD Enabler ensures secure delivery of DCD Content. The DCD Enabler also provides security on a per Channel basis with the same measures as used for the delivery mechanisms.

When the DCD Server delivers DCD Content utilizing broadcast bearers (e.g Cell Broadcast or MBMS) a notification describing any changes in subscription by a user at the client and the subsequent delivery of content associated to the change in subscription is sent by the client back to the authorized DCD Server.

4.2.3 Content Protection

The DCD Enabler can permit the DCD Content to be protected by utilizing [OMA-DRM].

4.2.4 Content and Subscription Information Privacy

The DCD Client has a responsibility to protect the privacy of, and restrict access to, DCD Content and subscription information contained within a DCD Client Device. A DCD Client Device supporting DCD-Enabled Client Applications will allow the subscriber identity to change over the life time of the terminal, either temporarily or permanently, whilst protecting DCD subscription information associated to previous subscriber entities. The DCD Enabler ensures the confidentiality of a user's DCD Content and subscription preferences. Functions to be considered include:

- Content Access Control due to Privacy: Based upon a user's privacy options, the DCD Client is required to deny access to DCD content on a terminal unless the user's identity (e.g. username / password) has been verified.
- Subscriber Validation: The DCD Client in conjunction with the DCD Server is required to verify a subscriber (e.g. via (U)SIM-based subscriber identity or user-supplied credentials) prior to accessing to a Service Provider's DCD supported services and associated content from the client.

4.3 Actors

In Figure 1, the DCD Content Providers and DCD-Enabled Client Applications are "actors" of the DCD Enabler. They are non-DCD entities via which users (Content Providers and end users) provide or consume services supported by the DCD Enabler.

Actors may use the external interfaces provided by the DCD Enabler (e.g. DCD-CPDE and DCD-CPR) thus allowing the deployment of DCD-based services using combinations of existing and new Content Providers and DCD-Enabled Client Applications.

4.3.1 DCD-Enabled Client Application

DCD-Enabled Client Applications are mobile-terminal-based applications, which can consume content delivered via the DCD Enabler, e.g. presenting the content to users, or using it for any arbitrary purpose. While the DCD Client must be aware of all DCD-Enabled Client Applications, the DCD-Enabled Client Applications may be aware or unaware of the DCD Enabler. For example, DCD-Enabled Client Applications may be designed to use the DCD Enabler in an implicit or explicit manner for one or more DCD Enabler functions, e.g.:

- The DCD-Enabled Client Application provides an Application Profile to the DCD Enabler, announcing its characteristics and the characteristics of the channels it seeks to access, and optionally using some or all of the content delivery facilities of the DCD Enabler, e.g.:
 - Making explicit content requests to the DCD Enabler and receiving responses
 - Asynchronously receiving delivery notifications or content
 - Simply consuming content that is delivered by the DCD Enabler to a pre-arranged content storage location
- The DCD-Enabled Client Application may simply consume content that is transparently made available by the DCD Enabler, e.g.:
 - The DCD Client may act as a local proxy (possibly transparent), e.g. through implementation-specific means supported by the terminal.
 - The DCD Client may be used to fill and manage a local content storage location (e.g. supported by the terminal filesystem) with content. The application may still request remote content, either directly from the local storage or via the DCD Client acting as a proxy as above.

As an example of the latter case, the DCD Enabler can provide content delivery to existing terminal browsers using the following approaches:

- The browser is configured to use the DCD Client as a local proxy.

- The browser is configured with a homepage or favorites which point to locally stored web pages that are maintained by the DCD Enabler. An arbitrary amount of local content can be stored and referenced, given that the browser supports local references, e.g. file://....

DCD-Enabled Client Applications interact with the DCD Enabler via:

- the DCD-CADE interface, for data exchange e.g. for requesting content and receiving delivered content.
- the DCD-CAR interface, for registration with the DCD Enabler.

DCD-Enabled Client Applications may be designed to embed the DCD Client as a library or a software component. When these two logical entities are combined in one physical entity:

- the DCD-CADE and DCD-CAR interfaces are no longer necessary, and are converted into internal functions of the DCD-Enabled Client Application
- The DCD-Enabled Client Application becomes responsible for compliance to the other requirements of a DCD Client

The behavior (e.g. rendering capabilities or interaction with other clients residing on a terminal) of the DCD-Enabled Client Application is out-of-scope of the DCD Enabler. No assumptions are made how any specific DCD-Enabled Client Application determines that additional content requests should be made through the DCD Enabler, if required. However, available methods may include:

- Via content references in the application content, using a DCD-related URI scheme; In this case the DCD-related URIs must be associated with the application, either by the DCD-Enabled Client Application or the terminal runtime component that maps URIs to target clients.

Note: Use of a DCD-specific URI scheme is an implementation-specific feature for DCD Release 1.0, and may be standardized in future releases.

- Via DCD Channel Metadata provided by the Content Provider or DCD Service Provider and accessible to the DCD-Enabled Client Application via the DCD Client, to define the rules for routing content references via the DCD Enabler.
- Via application-specific metadata exchanged between the Content Providers and DCD-Enabled Client Applications; While this metadata is opaque to the DCD Enabler, the DCD Enabler allows for its exchange along with application content. Such metadata may be used by Content Providers and DCD-Enabled Client Applications to define the rules for routing content references via the DCD Enabler, e.g.:
 - Associating specific URI domains to the DCD Enabler
 - Associating specific content types (e.g. as implied by filename “extensions” in URIs) to the DCD Enabler
 - Other application-specific methods

4.3.2 DCD Content Provider

DCD Content Providers interact with the DCD Enabler in an implicit or explicit manner for the purpose of utilizing one or more DCD Enabler functions, e.g.:

- The DCD Content Provider may directly serve requests for content, e.g. as any normal web server. In this case, the DCD Server appears as a normal web client.
- The DCD Content Provider may provide channel metadata which defines the characteristics of the channels it provides, and optionally support one or more content publication / delivery methods.
- The DCD Content Provider may publish available content to the DCD Server. The DCD Server will deliver such DCD Content to subscribed devices.

DCD Content Providers interacts with the DCD Enabler via:

- the DCD-CPDE interface, for data exchange e.g. for receiving content requests and requesting content delivery.
- the DCD-CPR interface, for registration with the DCD Server

5. Architectural Model

The DCD Enabler is realized as client- server architecture. It uses and interacts with other OMA enablers and non-OMA network functions or technologies. The proposed architecture is in line with the requirements in [OMA-DCD-RD].

5.1 Dependencies

The DCD Enabler can use other OMA Enablers and non-OMA functions or technologies.

- The DRM Enabler for content protection as described in [OMA-DRM-AD].
- The MCC Enabler for charging as described in [OMA-MCC-AD].
- The DM Enabler for the DCD-Enabled Client Application provisioning as described in [OMA-DM-AD].
- The PUSH Enabler for the service and system functions as well as content delivery, as described in [OMA-PUSH-AD] and [OMA-SIP_PUSH-AD].
- The Browsing Enabler for the service and system functions including content delivery as described in [OMA-Browsing].
- The BCAST Enabler for service administration and content delivery as described in [OMA-BCAST-AD].
- The CBCS Enabler for the content filtering as described in [OMA-CBCS-AD].
- The CSCF Enabler for the content screening as described in [OMA-CSCF-AD].
- The UAPROF Enabler for the adaptation of content based on terminal capabilities as described in [OMA-UAPROF].
- The DPE Enabler for the adaptation of content based on terminal capabilities as described in [OMA-DPE-RD].
- The 3GPP SMS for service administration and content delivery as described in [3GPP-SMS].
- The 3GPP Cell Broadcast (CBS) for service administration and content delivery as described in [3GPP-CBS].
- The Presence Enabler for the adaptation of content based on presence information as described in [OMA-PRS-IMPS-AD] and [OMA-PRS-SIMPLE-AD].
- The Location Enabler for the adaptation of content based on location information as described in [OMA-MLS-AD].

5.2 Architectural Diagram

Figure 2 describes the functional entities and interfaces of the DCD Enabler. The functional entities and interfaces in the DCD Enabler Architecture are described in depth in section 5.5.1.

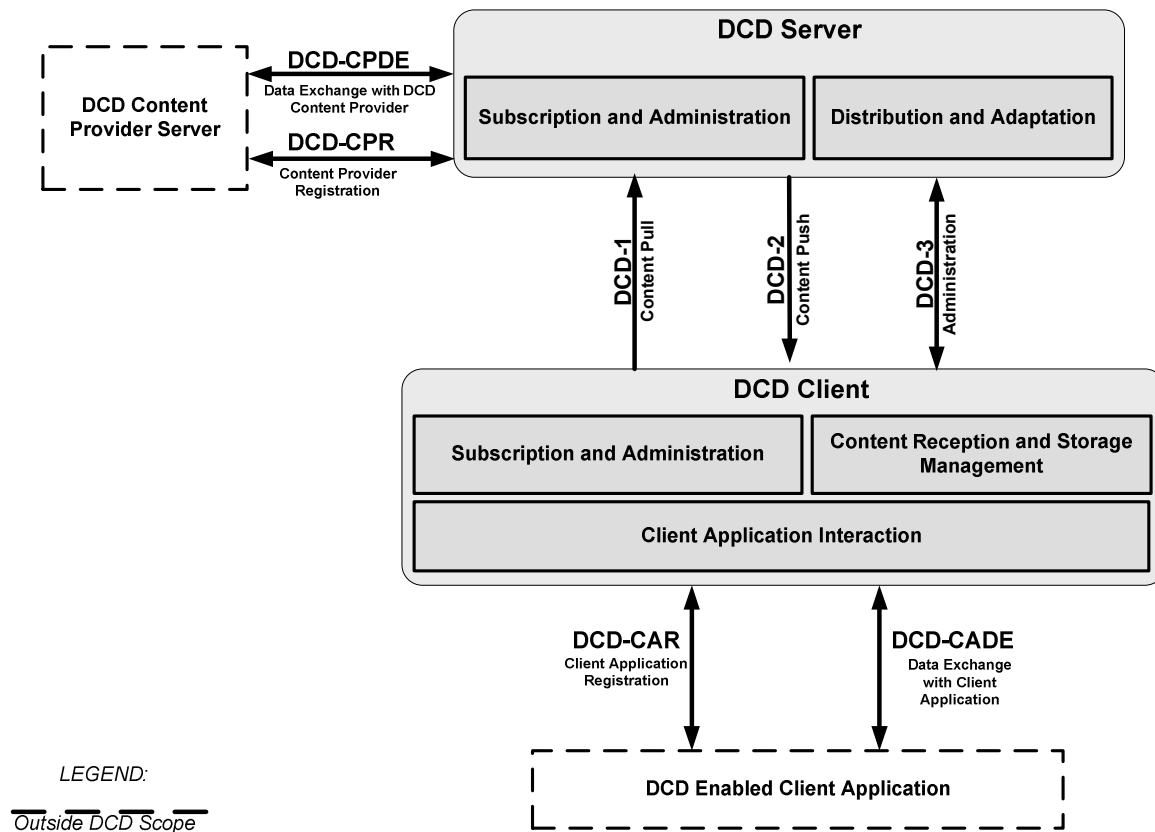


Figure 2: DCD Enabler Architecture

There are two main functional entities in the scope of the DCD Enabler:

- DCD Client resides in the mobile terminal and is used to communicate with the DCD Server. Three different logical functions can be differentiated inside this entity. The Subscription and Administration function (Client component), in charge of handling the exchange of service management information with the DCD Server. The Content Reception and Storage Management function, in charge of handling the content reception from the server. Finally, the Client Application Interaction Function that provides the necessary functions to make possible the interaction between DCD Enabler supported services and registered DCD Enabled Client Applications.
- DCD Server implements the application level network functionality for the DCD application. It is composed of two logical functions:
 - The Subscription and Administration function is in charge of handling the exchange of service management information between the DCD Server and DCD Client, and between the DCD Server and the Content Provider, such as when the Content Provider handles subscriptions.
 - The Distribution and Adaptation function distributes DCD Content and DCD Content notifications to the DCD Client.

The following entities are out of the scope, but may interact with the DCD Enabler:

- DCD Content Provider Server: interacts with the DCD Server in order to provide content and DCD Content and related information to the Enabler.
- DCD Enabled Client Application: interacts with the DCD Client in order to enable content delivery to the end user.

In order to maintain independence from the underlying implementation platform, all DCD interfaces need to be defined only in terms of data schema for the parameters of these interfaces. For the purpose of this document, the interface between two entities is called “bi-directional” if each entity offers a set of operations for use by the other entity. A uni-directional interface is an interface between two entities where only one entity offers a set of operations to the other entity. The bi-directional interface is a combination of two uni-directional interfaces. Each uni-directional interface could include synchronous (request-response) and / or asynchronous operations. The concept of uni-directional vs. bi-directional is an enabler interface level concept and independent of the underlying transport mechanism. The following interfaces are in scope of DCD:

- DCD-1 is a uni-directional point-to-point interface between the DCD Server and the DCD Client. This interface is used by the DCD Client to send content requests to the DCD Server and to receive DCD Content in the response. The DCD Client may include DCD Content received from a DCD-Enabled Client Application to be consumed by the DCD Content Provider in requests sent over this interface (e.g., subscription personalization information).
- DCD-2 is a uni-directional interface between the DCD Server and the DCD Client. This interface is used by the DCD Server to push notifications and / or content to the DCD Client. The DCD-2 interface could manifest itself as point-to-point push interface or point-to-multipoint broadcast interface.
- DCD-3 is a bi-directional interface between the DCD Server and the DCD Client. This interface is used by the DCD Server and the DCD Client to exchange DCD Subscription and Administration information. The DCD-3 interface could manifest itself as point-to-point interface or point-to-multipoint broadcast interface when applicable.
- DCD-CPR is a bi-directional interface between the DCD Content Provider and the DCD Server. This interface is used by the Content Provider to register new content channels with the DCD Server and to notify the DCD Server about subscription events and vice versa.
- DCD-CPDE is a bi-directional interface between the DCD Content Provider and the DCD Server. This interface is used by the Content Provider to publish content at the DCD Server and by the DCD Server to retrieve content from the Content Provider. The interface could also be used for exchange of administration information, if applicable. While the interface is bi-directional, only the DCD Server provided interface functions are a subject for standardization.
- DCD-CAR is a bi-directional interface between the DCD-Enabled Client Application and the DCD Client. This interface is used by the DCD-Enabled Client Application to register with the DCD Client when the application is installed on a terminal. In addition, this interface is used by DCD Client and DCD-Enabled Client Application to perform subscription related actions.
- DCD-CADE is a bi-directional interface between the DCD-Enabled Client Application and the DCD Client. This interface is used by the DCD Client to send notifications and / or content to the DCD-Enabled Client Application, and enables the DCD-Enabled Client Application to retrieve content from the DCD Client. The interface could also be used for exchange of administration information, if applicable. While the interface is bi-directional, only the DCD Client provided interface functions are a subject for standardization.

5.3 DCD Content Packaging

The DCD Content is packaged using a layered “envelope” model. Each envelope contains Content Metadata and the nested envelope. There is unique Content Metadata for each processing tier (i.e. the DCD Server, the DCD Client, and the DCD-Enabled Client Application), and each tier uses the appropriate Content Metadata to operate on the nested envelope prior to delivery to the next tier. Unlike Channel Metadata, which is associated with channel settings and provided upon registration of the DCD Enabled Client Application or the DCD Content Provider’s channels (see section 5.4), the Content Metadata is packaged and associated with the runtime DCD Content and passed from the DCD Server to the DCD Enabled Client Application through the DCD Client. If Channel Metadata is available, it is used by the processing tier prior to applying the Content Metadata.

Content payload inside the envelope is opaque to the DCD Enabler. The processing tier does not need the knowledge of nested envelope content but operates on processing tier using associated Content Metadata. However, the processing tier may use, create, or modify the Content Metadata for the next tier (e.g. the DCD Server may open the DCD Client Envelope and update priority setting in the Content Metadata for the DCD Client). In the common case scenario, the DCD Server envelope (including all nested envelopes) is provided by the DCD Content Provider. The Content Provider has intrinsic knowledge of the content, and therefore defines dynamic metadata (e.g. expiry, replacement, priority settings) associated with this content.

The envelope model for packaging the DCD Content ensures that the DCD Enabler operations are generic and independent from the content type.

The following is a brief description of the processing tiers, in addition to the related envelopes and Content Metadata. A schematic of the layered “envelope” model is provided in Figure #3.

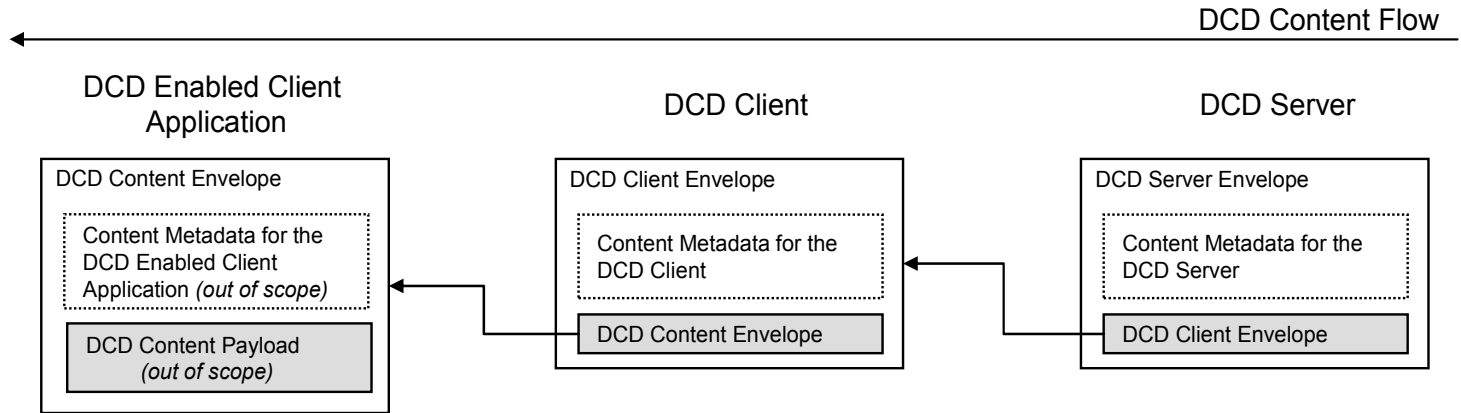


Figure 3: DCD Packaging – Envelope Model

1. The DCD Server: The DCD Server envelope contains the DCD Client envelope and Content Metadata for the DCD Server. This metadata is used by the DCD Server to handle the DCD Client envelope prior to delivery to the DCD Client.
2. The DCD Client: The DCD Client envelope contains the DCD Content envelope and Content Metadata for the DCD Client. This metadata is used by the DCD Client to handle the DCD Content envelope prior to delivery to the DCD-Enabled Client Application.
3. The DCD-Enabled Client Application: The DCD Content envelope contains the DCD Content payload. This content payload may have associated Content Metadata which the DCD-Enabled Client Application may use when processing the content. The content and any associated metadata are out of scope for the DCD.

5.4 Registration with the DCD Enabler

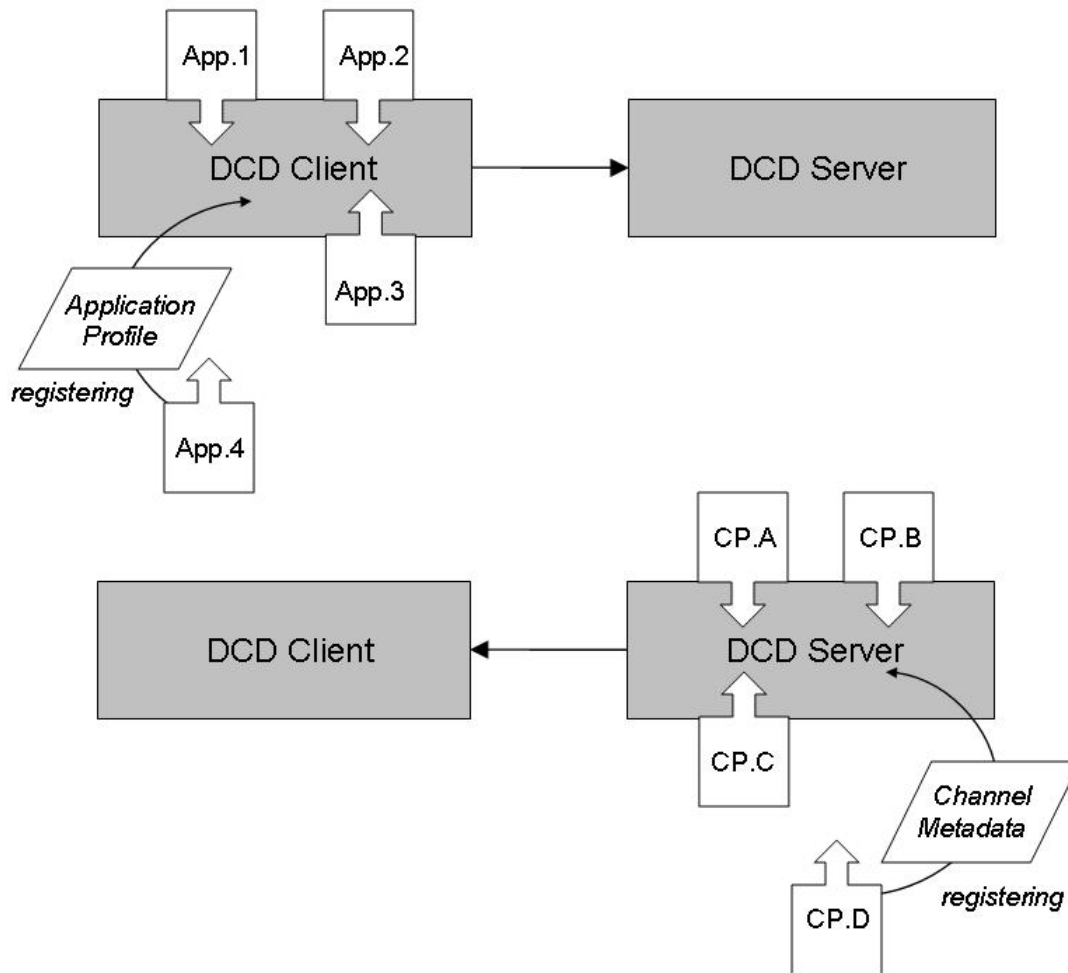


Figure 4: Registration with the DCD Enabler

Figure 4 shows the relationship between DCD Server and registered DCD Content Providers' channels, and the relationship between DCD Client and registered DCD Enabled Client Applications.

A DCD-Enabled Client Application registers with the DCD Enabler by interacting with the DCD-CAR interface of the DCD Client. The DCD-CAR interface allows dynamic registration of new applications. Upon registration, the DCD Enabled Client Application supplies its Application Profile (including Channel Metadata for the types of content that it supports). The Application Profile is not limited to DCD Client operations and may include the address or URL of the DCD Server associated with the application, and may also include settings for the DCD Server, if applicable and authorized.

Following the registration of the DCD-Enabled Client Application with the DCD Client, the DCD Client propagates the registration to a DCD Server. The DCD Client activation with the DCD Server, which results in session establishment, has to take place prior to registering application with the DCD Server. This registration may be achieved explicitly (e.g. by direct DCD Client and DCD Server interaction via DCD-3) or implicitly (e.g. by DCD Client using a broadcast transport enabler, which may not require interaction with a DCD Server).

In architectural terms, the process involving the registration of components of the DCD Enabler is understood as "activation" (i.e. activation of the DCD Enabler "pipe") and the process of registering external entities with the DCD Enabler is understood as "registration". The activation process may later reoccur, e.g. upon upgrade of the DCD Client or Server, or change in the DCD Server endpoint.

The DCD-Enabled Client Application may be preinstalled on the terminal and pre-registered with the DCD Client “out-of-the-box”. In this situation the address or URL of the DCD Server associated with the application may also be pre-defined.

Additionally, generic applications (i.e. “DCD unaware”) such as a browser or RSS feed viewer may be integrated into the DCD enabler. For example, by registration of the generic application using mediator entities that provide application profiles on behalf of the application, or using a local proxy entity for content delivery using DCD. Details of integrating such applications are out of scope of the DCD Enabler.

The DCD Content Provider registers its channels with the DCD Enabler by interacting with the DCD-CPR interface of the DCD Server. The DCD-CPR interface allows dynamic registration of new applications and content channels. Upon registration, the Content Provider supplies the Channel Metadata. The Channel Metadata supplied by a Content Provider is not limited to DCD Server operations, but may also include settings for a DCD Client.

Similarly to integrating generic applications with the DCD Client, a non-DCD Content Provider may interact with the DCD Enabler, for example, mediated registration performed through a DCD Service Provider or any other authorized entity.

Content delivery may start when both the DCD-Enabled Client Application and the appropriate channels are registered and bound together through the DCD Client and DCD Server. The binding between a DCD-Enabled Client Application and a DCD Content Provider is dynamic and may change according to the settings defined by the Application Profile. The DCD Enabler uses explicit or implicit binding mechanisms to bind registered DCD Enabled Client Application with registered DCD Content Providers’ channels.

In explicit binding, the association between DCD-Enabled Client Application and DCD Content Provider is achieved by matching DCD-Enabled Client Application ID with the URI of the Content Provider.

In implicit binding the association between DCD-Enabled Client Application and DCD Content Provider is achieved through the use of heuristics at the DCD Server to match the content types supported by the DCD-Enabled Client Application, as described by its Application Profile and identified by the DCD Enabled Client Application ID, with the content type attributes provided by the Content Provider during registration of the DCD Channels. The content type attributes are expressed in a standard format defined in [OMA-DCD-TS-Semantics]. The heuristics of implicit binding are implementation specific and out of scope of the DCD Enabler.

5.5 DCD Processes, Functional Components and Interfaces

5.5.1 DCD Processes

5.5.1.1 Activation

As described in section 5.4, the DCD Client needs to activate with the DCD Server. The activation process results in establishing the session. The activation of the DCD Client is a prerequisite to registration, subscription, and content delivery processes. The activation process may be triggered by device provisioning, installation of the DCD Client, registration of the first DCD Enabled Client Application, user request, etc. Client authentication may be required in order to proceed with the activation. The need for the use of authentication and the required authentication type are established based on Service Provider policy.

A single DCD Client may communicate with one or more DCD Servers. The DCD Server address may be preconfigured in the DCD Client, supplied by an application as a part of the Application Profile, or provided by other means (e.g. OMA DM).

5.5.1.2 Registration

As described in section 5.4, the DCD Enabled Client Application needs to register with the DCD Enabler in order to receive the DCD Content. At registration, the DCD-Enabled Client Application provides an Application Profile to the DCD Client that describes the application’s capabilities and preferences. The Application Profile represents channel metadata to facilitate channel offering for this application as well as application preferences common for all channels. Using DCD-CAR the DCD Enabled Client Application registers with the DCD Client and submits its Application Profile. The DCD Enabled Client Application registration and the submission of its Application profile could be mediated by another application submitting the Application Profile on its behalf.

Upon registration of the DCD Enabled Client Application, the DCD Client sends registration message to the DCD Server. Registration of applications with the DCD Server is required for point-to-point delivery models. Both the DCD Client activation and the application registration with the DCD Server may not be required for some broadcast delivery models.

The registration message to the DCD Server contains an entire Application Profile or a subset of the Application Profile. The DCD Server binds the DCD Enabled Client Application with one or more registered DCD Content Providers based on the preferences in the Application Profile. To facilitate bindings between the application and the content providers, the Application Profile may contain information about the content types supported by the application.

If the registration is successful, the DCD Server returns the list of content channels matching application preferences. This list is provided to the DCD Enabled Client Application with the registration response. For generic (DCD unaware) applications, which are registered indirectly, the response is provided to the mediator or omitted if the registration is internal to the DCD Enabler (e.g. the DCD Client is preconfigured to register the application).

5.5.1.3 Content Delivery

“Content delivery” as used in the context of this document includes the methods and overall process for establishing and maintaining the automated or on-demand delivery of content via the DCD Enabler. Content delivery is described here in terms of channel (the set of content items related to specific DCD-enabled services) lifecycle and operations. Further details on the following described functions are provided in the subsequent sections for the DCD functional entities and interfaces.

There are two main phases in the lifecycle of channels:

- Establishment: Channel establishment is the initiation of content delivery/reception of a channel by a specific DCD Client, and can be initiated by the DCD Server or DCD Client upon various events, including subscription actions occurring outside the DCD Enabler, or/and subscription requests by the DCD-Enabled Client Applications.
- Operations: the autonomous or on-demand delivery of content for the channel per the settings for the channel and its related content. Channels may be defined for various methods of delivery and periodic update.

The lifetime of channels and the applicable methods for channel delivery/update are determined by the DCD Service Provider and/or Content Provider for particular DCD-enabled services. Channel lifetime and the number of related content delivery transactions may vary, e.g. one request and a limited series of directly related responses (such as may be provided by a DCD-enabled browser proxy service), a limited-duration series of autonomous content updates (e.g. for sports events), or an indefinite series of autonomous content updates (e.g. a local weather channel).

DCD Client-initiated channel establishment begins with an initial request for content by the DCD Client via the DCD-1 interface, referencing a general or specific resource (e.g. the root content address of an application or channel, or any generic URI), and include various attributes of the requested resource as channel metadata. A DCD-Enabled Client Application uses the DCD-CAR interface to request channel establishment. The preferences for such channel establishment are identified by the Application Profile associated with the application. In its response, the DCD Server may include channel metadata which the DCD Client then uses to complete channel establishment, e.g. allocate content storage and schedule channel updates. Similarly, DCD Server-initiated channel establishment includes delivery of channel metadata which the DCD Client uses to complete channel establishment, e.g. setup content reception from the DCD-2 interface or DCD-1 interface.

Once the channel has been established, content delivery occurs via push (use of the DCD-2 interface for network-initiated content delivery), and/or pull (use of the DCD-1 interface for client-initiated content delivery) methods. Channel metadata may be updated as needed via the DCD-3 interface. Between the DCD Client and DCD-Enabled Client Applications, the DCD-CADE interface is used for content requests and delivery.

Content delivery may be automated and/or on-demand. Automated delivery may be controlled by scheduling DCD Client-initiated updates through channel metadata associated with the overall channel or individual content items, or DCD Server scheduling for pushed updates. Automated delivery may also be controlled by means outside the DCD Enabler, by Content Providers or DCD-Enabled Client Applications. From the DCD Enabler perspective, such externally-initiated delivery requests are handled the same as any on-demand delivery request by Content Providers or DCD-Enabled Client Applications.

If the DCD Server had previously requested confirmed delivery of the content, the DCD Client shall notify the DCD Server of content delivery and the DCD Server retries content delivery, based on DCD retry reschedule preferences or until the content is successfully delivered.

5.5.1.4 Personalization and Customization

As defined in [OMA-DCD-RD], personalization involves the dynamic adaptation of content according to a user's dynamic profile, whereas customization involves the static adaptation of content according to a user's preferences. As described in [OMA-DCD-RD], "content transformation" as a result of personalization and customization is out of scope of the DCD Enabler. Thus for the DCD Enabler, content adaptation is limited to controlling the way that content is delivered, i.e. what/when/how content is delivered based upon the dynamic profile and static preferences of the user and bearer selection for the delivery. For example DCD Content Provider may send different versions of content items (e.g. video files encoded for different quality and size for different bearers) to DCD Server in order to enable content adaptation and DCD Server may support delivery of different versions of the content items. The DCD Enabler may support external functions to "open and modify" content objects for the purpose of content transformation if required.

The scope of personalization and customization for the DCD Enabler includes the ability to control, e.g. the delivery of, DCD Content based upon a subscriber's profile attributes, which are provided by external enablers (e.g. presence and location), or managed by the DCD functional entities as data specifically related to DCD operation. The profile attributes specifically related to DCD operation include:

- statically applicable preferences (i.e. preferences in force until changed), usable for delivery customization (e.g. scheduling of DCD Channel delivery)
- dynamically applicable preferences (i.e. preferences based upon dynamically determined user profile attributes) , usable for application and content personalization

The following preferences as derived from [OMA-DCD-RD] are specified to provide a baseline of control options through which personalization and customization can be achieved:

- Per-channel static preferences
 - Subscription: This attribute is whether a subscriber is given access to a DCD channel, which is fulfilled by filtering received DCD channels at the client in the broadcast case
 - Delivery schedule and content expiry: These attributes control, for example, the periodic update, and the time-to-refresh content items, and the scheduling of the delivery of content items according to the time of day.
 - Storage space usage: This attribute represents the minimum / maximum content storage space for each DCD channel
 - Manual suspension: This attribute indicates whether delivery for a DCD channel is currently manually suspended
- Per-channel dynamic preferences
 - Deliver when roaming: This attribute conditions the delivery (i.e. deliver or do not deliver) of content channels based upon a subscriber's roaming status
 - Deliver based upon current location: This attribute conditions the delivery of content channels based upon a subscriber's location.
 - Deliver based upon Presence attribute: This attribute conditions the delivery of content channels based upon the value of a specific attribute value provided by a Presence Service.

These preferences are under the control of the subscriber and the DCD Service Provider. User (or subscriber) preference control is governed by the DCD Service Provider's global policies for the DCD channel, including policies expressed by the Content Provider. In some cases a subscriber is not provided with control over specific preferences.

The methods by which the preferences are managed by the DCD Service Provider or subscriber are out of scope of the DCD Enabler.

The role of the DCD Server and DCD Client in controlling content delivery based on the specified preferences is described in [OMA-DCD-TS-Semantics].

5.5.2 DCD Functional Entities

5.5.2.1 DCD Client

The DCD Client resides on the terminal and interacts with DCD server and the DCD Enabled Client Application.

The DCD Client provides the functions as described in the subsequent sections.

5.5.2.1.1 Subscription and Administration Function

The Subscription and Administration function has the responsibility for the following actions:

- Activation of the DCD Client with the DCD Server
 - The activation of the DCD Client is a prerequisite to registration, subscription, and content delivery. The activation results in session establishment. Client authentication may be required in order to proceed with the activation.
- Registration of the DCD Enabled Client Application with the DCD Enabler
 - DCD Enabled Client Application registers with the DCD Enabler in order to receive DCD Content. Upon registration, the DCD Enabled Client Application provides an Application Profile (including Application-ID). The DCD Client forwards the registration information (i.e. the Application Profile or its subset) to the DCD Server.
- Deregistration of the DCD Enabled Client Application
 - Deregistration process occurs when the DCD Enabled Client Application initiates deregistration request to the DCD Client. Such request could be issued as a result of user action, software removal, etc. Alternatively, the DCD Client may deregister an application upon discovering that the application is no longer available.
- Deactivation of the DCD Client
 - Upon deregistration of the last DCD Enabled Client Application or the change of device user, the DCD Client may deactivate with the DCD Server. The deactivation results in session removal.
- Session management
 - The session describes the context of DCD content delivery. In the point-to-point delivery a session is a continuous logical association between the DCD Server and the DCD Client established for the purpose of exchanging the DCD Content. In the broadcast delivery a session represents logical association between the DCD Server and the group of DCD Clients. The session is removed upon DCD Client deactivation.
- Reporting of security violations (security-related errors) and statistics
 - In accordance with the DCD Service Provider policy, the DCD Client may report security violations (security-related errors) and usage statistics to the DCD Server.
- Discovery of available DCD Channels
 - After registration, the DCD-Enabled Client Application may obtain the list of available DCD Channels matching Application Profile preferences. This list contains relevant subset of the Channel Metadata allowing the application to decide whether or not to subscribe to the available channels.
- Subscription handling
 - The DCD Enabled Client Application may request subscription to DCD Channels. The DCD Client handles these subscriptions. The DCD Client notifies the DCD Server about the subscriptions in point-to-point case and may notify in some broadcast cases. In broadcast case, the DCD Client supports

content filtering. DCD Client also facilitates validation of subscription requests coming from the DCD Server (for subscriptions established outside the DCD Enabler).

- DCD channel suspension and resumption on behalf of the DCD Service Provider
 - The DCD Client handles channel suspension and resumption requests from the DCD Server. These requests may be originated by the DCD Content Provider or the DCD Service Provider.
- DCD channel suspension and resumption on behalf of the DCD Enabled Client Application
 - The DCD Client handles channel suspension and resumption requests from the DCD Enabled Client Application and notifies DCD Server upon such requests

5.5.2.1.2 Content Reception and Storage Management Function:

The Content Reception and Storage Management Function has the responsibility for the following actions

- Reception of DCD Content via pull, push, or broadcast methods
 - The DCD Client receives DCD Content from the DCD Server. The content is delivered over DCD-1 or DCD-2 interfaces.
- Handling of connection parameters such as:
 - DCD interface to use for content delivery, i.e. DCD-1 and/or DCD-2.
 - DCD Server and/or proxy address and data connection details, for requests made via DCD-1 and DCD-3: the IP-routable address of the DCD Server, e.g. as a domain name or IP address. These may be individually specified for DCD-1 and DCD-3.
 - Connection security to be applied to requests made via DCD-1 and DCD-3: this defines whether a non-secure (e.g. HTTP) or secure connection (e.g. HTTPS) must be used.
 - Transport method, for content delivered via DCD-2: this indicates from which terminal the DCD Client should expect content delivered via DCD-2, e.g. WAP Push client, BCAST client, or SMS client (for Cell Broadcast).
 - Broadcast channel access parameters, for content delivered via DCD-2: parameters specific to the broadcast transport in use, e.g. for Cell Broadcast, the channel number.
- Processing of DCD Content according to settings defined in the DCD content metadata
 - Upon reception of the DCD Content, the DCD Client executes storage management functions according to local policies (e.g. to ensure adequate storage is available) and according to the settings defined in content metadata (e.g. removal of expired content, replacement of outdated content, priority handling, etc.)
- Specific content delivery functions:
 - Support for content bundles and channel aggregation for bandwidth efficiency
 - Support for content delivery confirmation, if requested by the DCD Server
 - When requested by the DCD Server, the DCD Client notifies the DCD Server following the successful or unsuccessful delivery of DCD Content.
- Content storage management
 - The DCD Client manages content storage according to the preferences defined by the DCD-Enabled Client Application at registration
 - The DCD-Enabled Client Application can request specific content storage options or actions from the DCD Client, such as:
 - Reserve space, retrieve, or erase (part of or all) the DCD content in the DCD Client-controlled content storage.

- Provide DCD Client with access to the content storage controlled by the DCD-Enabled Client Application for content update, when required
- Set DCD Content storage and content retention policies for a particular DCD channel.

5.5.2.1.3 Client-Application Interaction Function

The Client-Application Interaction Function provides the following functions to the DCD Enabled Client Applications:

- Support for registration with a DCD Client
 - The DCD Client uses Client-Application Interaction Function to register the DCD Enabled Client Application. During registration, the application submits an Application Profile.
- Support for deregistration with a DCD Client
 - The DCD Client uses Client-Application Interaction Function to deregister the DCD Enabled Client Application.
- Support for subscription to DCD Channels and subscription for broadcast channels
 - Channel selection. After registration, the DCD-Enabled Client Application interacts with Client-Application Interaction Function to select DCD Channels according to the channel metadata provided by the Content Providers at registration.
 - Notification of new DCD Channel availability. Using Client-Application Interaction Function the DCD Client may notify the DCD Enabled Client Application about availability of new DCD Channels matching application preferences.
 - Subscription for broadcast DCD Channels. The DCD Client filters broadcast channels according to application defined channel selection. The Client-Application Interaction Function makes possible for the DCD Enabled Client Application to indicate its interest in specified broadcast DCD Channel.
- Support for content delivery
 - Notification of new content delivery via DCD. Client-Application Interaction Function allows DCD-Enabled Client Applications to receive notifications that the new content has been delivered by the DCD Enabler, and is available in the terminal's storage.
 - Requests for DCD Content and associated responses. Client-Application Interaction Function allows DCD-Enabled Client Applications to request content update and to receive responses.
 - Reception of asynchronously delivered content. Client-Application Interaction Function allows DCD-Enabled Client Applications to receive new content that has been delivered asynchronously (Push or Broadcast).
 - Notification of DCD Channel suspension and resumption. This allows a DCD Enabled Client Application to receive notifications that the DCD Channel has been suspended. The DCD Enabled Client Application may use this notification for arbitrary purposes, e.g. to update graphical UI elements (e.g. a "connection status" icon, or graying out of menu options for manual updates) or suspend requests for synchronous DCD Content updates.
 - Requests for DCD Channel suspension and resumption. This allows a DCD-Enabled Client Application to request DCD Channel suspension and resumption. The DCD Client notifies the DCD Server upon such request.
- Application Profile handling
 - This function allows the DCD Client to handle delivery of the DCD Content for a particular DCD Enabled Client Application according to its Application Profile.

5.5.2.2 DCD Server

The DCD Server implements the network side of the DCD Enabler and provides the functions as described in the subsequent sections.

5.5.2.2.1 Subscription and Administration Function

The Subscription and Administration function has the responsibility for the following actions

- DCD Content Providers registration of channels with the DCD Enabler
 - DCD Content Provider registers its channels with the DCD Server. Upon registration, the DCD Content Provider supplies Channel Metadata for supported DCD Channels. The DCD Content Provider registers additional channels when they become available and updates registration for existing channels when the channel preferences change.
- Deregistration of the DCD Channels by DCD Content Providers
 - Channel deregistration process occurs when the DCD Content Provider terminates offering the content for this DCD Channel.
- Subscription event notifications by DCD Content Providers
 - In case when DCD Content Provider handles subscription events (e.g. subscriptions and unsubscriptions), it informs DCD Server about the events via the DCD-CPR interface. If required, it is the DCD Enabler's responsibility to validate external subscriptions with the DCD Enabled Client Applications.
- Activation of the DCD Client
 - Upon activation, the DCD Client and the DCD Server establish a new session. This session provides the context for content delivery and administrative actions. The DCD Server may need to authenticate the DCD Client in order to process activation.
- Registration of the DCD Enabled Client Application
 - Upon registration of the DCD Enabled Client Application, the DCD Client forwards registration information to the DCD Server. The DCD Server receives the Application Profile (or its subset) and binds the DCD Enabled Client Application with the DCD Channels matching application preferences. The list of matching channels is returned with the registration response.
- Deregistration of the DCD Enabled Client Application
 - Upon deregistration of the DCD Enabled Client Application, the DCD Client deregisters it with the DCD Server. The deregistration results in termination of DCD content delivery to this DCD Enabled Client Application, and may trigger DCD Client deactivation. The DCD Server may also notify Content Providers of deregistration of the DCD Enabled Client Application, if requested by the Content Providers.
- Deactivation of the DCD Client
 - Upon deregistration of the last DCD Enabled Client Application or the change of device user, the DCD Client may deactivate with the DCD Server. The deactivation results in session removal.
- DCD Client subscription event handling
 - The DCD Server handles subscription events from DCD Client via the DCD-3 interface.
- Collection of security violations (security-related errors) and statistics from the DCD Client
 - The DCD Client may report security violations (security-related errors) and usage statistics to the DCD Server, if required.
- Session management

- Upon activation, the DCD Client and the DCD Server establish a session. The session provides the scope for content delivery between the DCD Client and the DCD Server. The DCD Server determines when to close the session or may do so upon DCD Client request.
- DCD channel suspension and resumption by DCD Service Provider or on behalf of the DCD Content Provider
 - The DCD Server handles channel suspension and resumption requests from the DCD Content Provider. These requests are either explicit (i.e. over DCD-CPDE interface) or implicit (e.g. as defined in Channel Metadata submitted by the Content Provider), or as required by DCD Service Provider). DCD Server acceptance of suspension and resumption actions is performed upon an authorized request and in accordance with the DCD Service Provider's policy.
- DCD channel suspension and resumption on behalf of the DCD Enabled Client Application
 - The DCD Server handles channel suspension and resumption requests from the DCD Client. These requests are either explicit (i.e. over DCD-3 interface) or implicit (i.e. as defined in Channel Metadata submitted by the DCD Enabled Client Application).

5.5.2.2 Content Distribution and Adaptation Function

The Content Distribution and Adaptation function has the responsibility for the following actions

- Delivery of DCD Content via pull, push, or broadcast methods
 - The DCD Server receives content from the DCD Content Provider and delivers it to the DCD Client over DCD-1 or DCD-2 interfaces. The DCD Content is delivered to the DCD Client on demand, on schedule, on application request, or upon availability at the DCD Server.
- Processing of DCD Content according to the Content Metadata
 - The DCD Server processes DCD Content according to the settings defined in content metadata (e.g. removal of expired content, replacement of outdated content, priority delivery, fragmentation due to size limitations, etc.)
- Specific content distribution functions:
 - Support for content publishing by the DCD Content Providers
 - Support for channel bundles e.g. delivering updates for multiple channels in a single response action for bandwidth efficiency
 - Facilitate content adaptation according to terminal characteristics
 - Facilitate content personalization according to presence and location information
 - Support for content delivery according to settings and retrieval scheduling as defined by Channel Metadata
 - Support for handling content delivery confirmation
 - When required, the DCD Server requests from the target DCD Client the successful or unsuccessful delivery of DCD Content.
- Content storage management
 - The DCD Server allocates and manages temporary storage for content in delivery. This storage is used for the DCD Content:
 - Pending delivery to "out-of-coverage" terminals
 - Pending delivery confirmations
 - Fragmented to satisfy limitations of the terminal's storage
 - Pending delivery slot as per the delivery schedule

5.5.3 External Enabler Dependencies

The following sections describe the external OMA Enabler entities with which the DCD Enabler interacts directly in the fulfillment of core DCD requirements, e.g. in the operation of DCD-specified interfaces.

Except where noted, the external enabler entities are not mandatory dependencies of the DCD Enabler in any specific deployment, e.g. the DCD Enabler can be used when only some of the external enabler clients referenced here are present in a terminal. What is mandatory is that the DCD Enabler be capable of interfacing (if applicable) to these external enablers, when they are present in any specific deployment.

Where the DCD Enabler interacts directly with external OMA enablers, it does so under the enabler exposure control environment called for in [OSE]. Depending upon the specific policies applicable between the DCD Service Provider and the provider of the external enablers, a policy enforcement step may be required.

5.5.3.1 Push Enabler Entities

This section describes Push Enabler functions and their related functional components for reuse by the DCD Enabler. The decision on whether some of these Push functional components are implemented in DCD server / client is made at the time of DCD service deployment.

The DCD Server uses the Push Enabler for delivery of content or administration actions to the DCD Client either by interacting with:

- the Push Proxy Gateway (PPG) entity via the Push Access Protocol (PAP) interface defined in [OMA-PUSH-PPG] for adaptation of Push operations to underlying bearers.
- the Push Client entity via the Push Over-the-Air protocol (Push-OTA) interface defined in [OMA-PUSH-OTA].
- The SIP Push Client entity via the Push over SIP protocol interface defined in [OMA-SIP_PUSH]

The DCD Client uses the Push Enabler for reception of content or administration actions from the Push Client. The interface between the Push Client and DCD Client is implementation-dependent and unspecified by OMA.

Push messages received by the Push Client are routed to the DCD Client via association with an OMNA-assigned Push-OTA Application-ID.

5.5.3.2 BCAST Enabler

This section describes BCAST Enabler functions and their related functional components for reuse by the DCD Enabler. The decision on whether some of these BCAST functional components are implemented in DCD server / client or BCAST server / client, is made at the time of DCD service deployment.

To enable the use of any of the BCAST functions listed in this section for broadcast DCD content, it is presumed that the DCD Enabler supports the use of the BCAST entities and BCAST functional components relevant to each specific function, as defined in [OMA-BCAST-AD].

The DCD Content has to be agnostic for the BCAST Enabler, hence the DCD Server will use the BCAST-1 reference point as defined in [BCAST-AD] for interaction with the BCAST enabler. For the interaction between both enablers in the Client side, there isn't a reference point exposed in [BCAST-AD], thus it will be implementation dependent.

Point-to-point DCD supported applications may also make use of some of the BCAST entities and BCAST functional components defined in [OMA-BCAST-AD] for each of the BCAST functions listed in this section (e.g. administration purposes, personalization of BCAST Services etc.), together with the relevant mechanisms specified in DCD.

DCD Enabler's use of BCAST Enabler is based on some or all of the following BCAST functions:

- Service Guide Function: BCAST Service Guide can be used to provide users with information about delivered DCD Content.

- File / Stream Distribution Function: BCAST File and Stream Distribution Functions can allow respectively efficient file and streaming delivery for the broadcast of DCD contents to DCD Clients.
- Service and Content Protection Function
 - BCAST Service Protection enables access control for BCAST channels via which DCD Content can be delivered. Use of the channel as a delivery medium for DCD Content presumes that DCD Client host terminals have access to the related BCAST channels. This may require pre-distribution of BCAST channel access rights to DCD -enabled terminals, or dynamic distribution of BCAST channel access rights if these are affected by user or Service Provider subscription actions.
 - BCAST Content Protection can enable access control for individual broadcasted DCD Content, once DCD Content is in the terminal. This may require pre-distribution of BCAST content access rights to DCD-enabled terminals, or dynamic distribution of BCAST content access rights if these are affected by DCD Service / content subscription actions.
- Service Provisioning Function: The BCAST Service Provisioning Function can enable user subscription to DCD supported services and Content, and charging related functions. This involves point to point channel capabilities.
- Notification Function: The BCAST Notification function can allow the efficient delivery of DCD notifications, which can be used to inform the DCD Client about upcoming events related to the DCD supported services or Content.

5.5.4 Interfaces

In order to maintain independence from the underlying implementation platform, all DCD interfaces need to be defined only in terms of data schema for the parameters of these interfaces.

5.5.4.1 Interface DCD-1

DCD-1 interface is a uni-directional point-to-point interface between the DCD Server and the DCD Client. Via this interface the DCD Client sends content pull requests to the DCD Server, and receives responses. The DCD Client sends requests to the DCD Server via DCD-1 interface as a result of several triggers, e.g.:

- Automatic request upon notification of content availability at the DCD Server
- Automatic request for content upon a predefined schedule
- Automatic request for content upon stored content expiration
- On-demand request for content by the DCD-Enabled Client Application, e.g. upon end-user interaction or application-initiated interaction

The DCD Client may include DCD Content received from a DCD Enabled Client Application to be consumed by the DCD Content Provider in requests sent over this interface (e.g. subscription personalization information).

5.5.4.2 Interface DCD-2

DCD-2 interface is uni-directional interface between the DCD Server and the DCD Client. The DCD-2 interface is supported as a point-to-point content push interface and point-to-multi-point broadcast interface. Via this interface the DCD Server delivers pushed notifications and / or content to the DCD Client, e.g.

- Notification of content availability for client-invoked retrieval via DCD-1 interface
- Direct delivery of content

As DCD-2 interface is supported as a point-to-point content push interface and point-to-multi-point broadcast interface, the semantics of notifications indicating content availability differ according to whether broadcast or point-to-point modes are used. In the point-to-point case, these notifications are targeted to a specific application and terminal user, and the message contains application ID, terminal ID, etc. In the broadcast case, the DCD Server expects the DCD Client to filter these notifications (e.g. based on the application ID and / or content type match with registered DCD Enabled Client Applications).

The use of this interface is initiated by the DCD Server upon request by a Service Provider / Content Provider (through DCD-CPDE interface), or by the DCD Server itself.

5.5.4.3 Interface DCD-3

DCD-3 interface is a bi-directional interface between the DCD Server and the DCD Client. The DCD-3 interface is supported as a point-to-point interface and as point-to-multipoint broadcast interface when applicable. Via this interface the DCD Client and the DCD Server perform channel subscription and administration functions, e.g.:

- DCD Client Activation and session management
- DCD-Enabled Client Application registration actions
 - Register
 - Deregister
- Service administration actions, e.g.
 - change channel delivery options
 - suspend / resume channel
- Channel Discovery actions
- Channel subscription actions
 - subscribe / unsubscribe
 - subscription validation for external subscriptions

The use of this interface can be initiated by:

- DCD Client upon request of a DCD-Enabled Client Application (through DCD-CAR interface)
- DCD Client (e.g. activation / deactivation of DCD Client, application registration / deregistration)
- DCD Server (e.g. channel un-subscription or new channel announcement over push or broadcast bearers)

5.5.4.4 Interface DCD-CPR

DCD-CPR is a bi-directional interface between the Content Provider and the DCD Server. This interface is used by the Content Provider to register new content channels with the DCD Server. The Content Provider uses DCD-CPR interface when the new channels become available or when there is an update to the settings of the existing channels. The interface is used to supply Channel Metadata to the DCD Enabler. The DCD-CPR interface allows the plug-in of new Content Providers' channels into the DCD Server dynamically i.e. there is no need for the DCD Server restart or interruption of DCD Server operations upon registration. The registration of the Content Providers' channels is independent from the registration of the DCD Enabled Client Applications. This interface is also used to exchange subscription related information between the DCD Content Provider and the DCD Server. Content Provider notifies the DCD Server about subscription events. In addition, the DCD Server may use this interface to notify the DCD Content Provider about subscription events from a DCD Client.

5.5.4.5 Interface DCD-CAR

DCD-CAR is a bi-directional interface between the DCD-Enabled Client Application and the DCD Client. This interface is used by the DCD-Enabled Client Application to register / deregister with the DCD Client when the application is installed on a terminal or upgraded and to perform subscription related actions (e.g. subscribe and unsubscribe). The DCD-Enabled Client Application uses DCD-CAR interface to supply Application Profile (including the Channel Metadata) to the DCD Enabler. The DCD-CAR interface allows the plug-in of new DCD-Enabled Client Applications into the DCD Enabler dynamically i.e. there is no need for the DCD Client restart or interruption of DCD Client operations upon registration. The registration of the DCD-Enabled Client Applications is independent from the registration of the Content Providers' channels. In addition, DCD Client may use this interface to verify external subscriptions with DCD-Enabled Client Application.

5.5.4.6 Interface DCD-CPDE

DCD-CPDE is a bi-directional interface between the Content Provider and the DCD Server. This interface is used by the Content Provider to publish content at the DCD Server and for the DCD Server to retrieve content from the Content Provider. The interface DCD-CPDE is used to exchange data packaged according to the envelope model described in section 5.3. The data inside the envelope is opaque to the enabler, therefore only the structural parameters of the DCD Server envelope need

to be specified. Content publication by a Content Provider must conform to the interface offered by the DCD Enabler. Requests for content retrieval by the DCD Server must conform to the interface offered by the Content Provider, and may include metadata (e.g. HTTP headers) useable by the Content Provider to optimize / structure content for DCD delivery.

5.5.4.7 Interface DCD-CADE

DCD-CADE is a bi-directional interface between the DCD-Enabled Client Application and the DCD Client. This interface is used by the DCD Client to send notifications and / or content to the DCD-Enabled Client Application and by the DCD-Enabled Client Application to retrieve updated content from the DCD Client. The interface DCD-CADE is used to exchange data packaged according to the envelope model described in section 5.3. The data inside the envelope is opaque to the enabler, therefore only the structural parameters of the DCD Content envelope need to be specified. The mechanism for DCD Client interaction with the DCD Enabled Client Application is platform specific. The data schema for this interface is defined in the Application Profile, provided during DCD-Enabled Client Application registration.

5.6 Flows

The legend used in this section is as follows:



5.6.1 Registration of the DCD Enabled Client Application

There are two distinct examples for registration of the DCD Enabled Client Application.

Example 1: Application is aware of the DCD Content Provider address prior to the registration.

The DCD Enabled Client Application may be aware of the DCD Content Provider address prior to registration with the DCD Enabler. This information could be provided to the DCD Enabled Client Application as a result of DCD Content discovery by external means, provided through direct interaction with the terminal user, predefined by the application vendor, or setup by the service provider (e.g. using OMA DM).

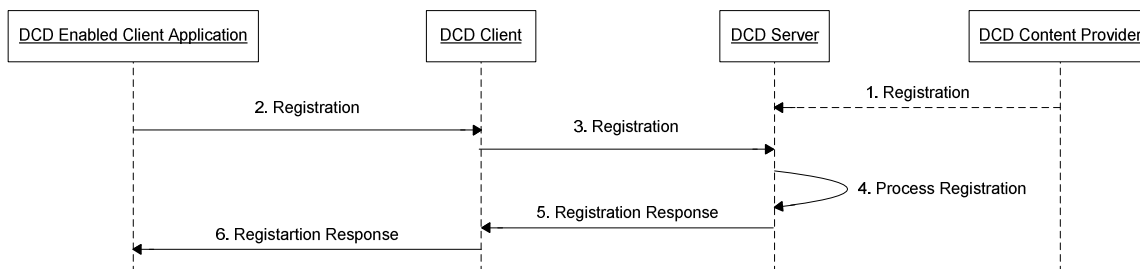


Figure 5: Registration of DCD Enabled Client Application

1. DCD Content Provider registers content channels with the DCD Server.
2. DCD Enabled Client Application registers with the DCD Client providing the Application Profile (AP). The AP contains the DCD Content Provider address and other settings specifying application capabilities and preferences.
3. DCD Client sends registration message to the DCD Server. The message contains the entire AP or the server-side applicable subset of the AP. In this example, the DCD Server address must be known *a priori*, or determined by other implementation-specific means.
4. DCD Server validates the AP and binds the application with the specified Content Provider and with the channels offered by this Content Provider. The registration process may include user authorization for access to the requested

DCD Content. The DCD Server may also interact with the Content Provider through non-DCD interfaces at this time (e.g. for authorization or service provisioning)

5. DCD Server sends an Registration Response. If the registration was successful, the registration response message will contain channel metadata for the channels associated with the registered application.
6. DCD Client returns a Registration Response to the application. If the registration was successful, the application is now ready to initiate pull-based content delivery via the DCD-1 interface or to receive pushed content via the DCD-2 interface. The application may need to subscribe to the requested content channel with the DCD Content Provider, if applicable. Alternatively, the subscription may occur outside of the DCD Enabler.

Example 2: Application is unaware of DCD Content Provider address at registration.

The DCD Enabled Client Application provides its preferences and capabilities including supported channel types (e.g. MIME types) to the DCD Client. It expects the DCD Enabler to discover content channels matching application preferences and provide this information upon registration. Examples of such applications are general purpose, off-the-shelf applications designed to handle specific content types (e.g. RSS Feed Viewer, Video Player, etc.), or applications designed for a specific business domain (e.g. Stock Quotes, Traffic News, etc.)

Refer to Figure 5:

1. DCD Content Provider registers content channels with the DCD Server.
2. DCD Enabled Client Application registers with the DCD Client providing the Application Profile (AP). The AP contains application capabilities and preferences including content types for channels supported by this application.
3. DCD Client sends an application registration message to the DCD Server. The message contains the AP or an applicable subset of the AP. In this example, the DCD Server address must be known *a priori*, or determined by other implementation-specific means.
4. DCD Server validates the AP and matches its content type preferences with available content channels registered by Content Providers. It constructs a list of channels matching AP preferences. The registration process may include user authorization for access to the requested DCD Content. The authorization may limit the list of channels offered to the application. The DCD Server may also interact with applicable Content Providers through non-DCD interfaces at this time (e.g. for authorization or service provisioning).
5. DCD Server sends a Registration Response. If the registration is successful, the message contains the list of applicable channels, and associated channel metadata.
6. DCD Client returns a Registration Response to the application. If the registration was successful, the response may contain the list of channels matching AP preferences. Alternatively, the application may retrieve the list of channels from the DCD Client. The application chooses desired content channels. It may need to subscribe to selected content with the DCD Content Providers, if applicable. The subscription with the DCD Content Providers is out-of-scope for the DCD Enabler, but could be treated as an opaque data exchange over the DCD-1 interface. Alternatively, the subscription may occur outside of the DCD Enabler.

5.6.2 Registration of the DCD Enabled Client Application without interaction with the DCD Server

This example is applicable to certain broadcast scenarios such as receive-only broadcast kiosk or emergency broadcast. In these scenarios the registration is local to the terminal and does not result in registration information being sent to the DCD Server.

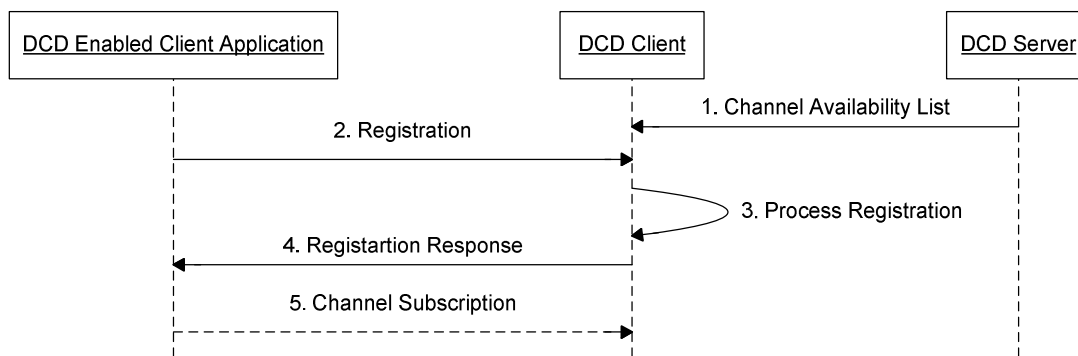


Figure 6: Registration of the DCD Enabled Client Application without DCD Client interaction with the DCD Server

1. DCD Server broadcasts a number of available channels and associated metadata
2. DCD Enabled Client Application registers with the DCD Client providing the Application Profile (AP). The AP contains application capabilities and preferences including content types for channels supported by this application.
3. DCD Client validates the AP and matches its content type preferences with available broadcasted content channels. Based on AP settings and channel metadata provided by the DCD Server at step 1, the DCD Client auto-subscribes the application to certain broadcast channels (e.g. emergency alerts, advertisement, etc.). It also prepares a list of “optional” channels matching AP preferences.
4. DCD Client returns a Registration Response to the application. It may contain the lists of auto-subscribed channels and of “optional” broadcast channels matching AP preferences. Alternatively, the application may retrieve these lists from the DCD Client.
5. The application chooses desired broadcast channels from the “optional” list. It issues a Channel Subscription to notify the DCD Client of selected “optional” channels, if any. The DCD Client starts monitoring the selected broadcast channels for new content.

5.6.3 Channel Subscription

This flow illustrates “top-down” channel subscription, i.e. when the DCD-Enabled Client Application sends channel subscription request to the DCD Server. There are two distinct scenarios for channel discovery leading to such subscription.

1. Internal channel discovery upon registration response or upon channel availability notification

At registration, the DCD-Enabled Client Application receives the metadata for the channels matching application preferences. Additionally, an already registered application may receive the metadata for new matching channels when they become available. The subset of Channel Metadata exposed to the DCD-Enabled Client Application contains channel name, description, identifier, etc. Based on this information, the application selects a set of DCD Channels to subscribe.

2. External channel discovery (e.g. via browser)

Users may discover DCD channels through subscription portals external to the DCD Enabler. The subscription portal may be associated with the DCD Service Provider or a DCD Content Provider. In order to complete the subscription over the DCD Enabler, the DCD-Enabled Client Application needs to obtain the identifier for a selected channel. The mechanism for communicating channel identifier from a discovery agent to a DCD-Enabled Client Application is device and application specific and out of scope for DCD. Alternatively, the external channel discovery may trigger registration of the DCD-Enabled Client Application and the identifier for a selected channel may be specified as a part of metadata included in the Application Profile.

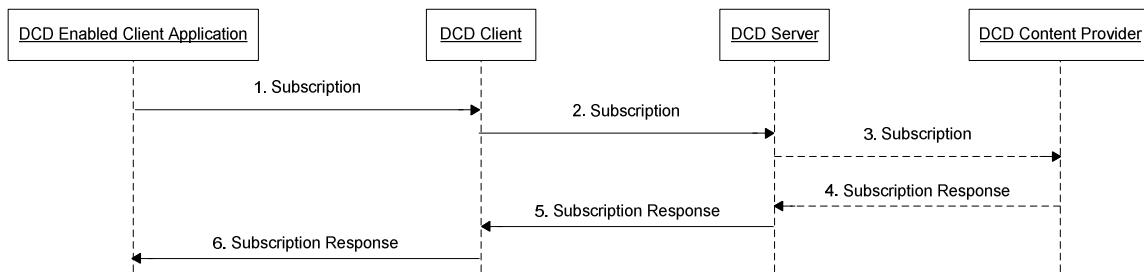


Figure 7: Channel Subscription by a DCD-Enabled Client Application

1. DCD-Enabled Client Application sends subscription request to the DCD Client specifying channel identifier for a selected DCD Channel. This channel identifier corresponds to the channel already registered with the DCD Server.
2. DCD Client sends subscription request message to the DCD Server.
3. DCD Server validates subscription request and optionally forwards the subscription to the DCD Content Provider, which offers the channel. The subscription message to the DCD Content Provider contains information that associates the user and / or device with the subscription.
4. DCD Content Provider validates the subscription and sends subscription response.
5. DCD Server sends subscription response to the DCD Client and establishes the subscription, if successful.
6. DCD Client returns subscription response to the application. If the subscription was successful, the application is now ready to receive the content for a selected channel or to establish subscription personalization settings with the DCD Content Provider, where applicable.

5.6.4 Notification on External Channel Subscription

In this example the DCD Content Provider notifies the DCD Server about channel subscription established outside the DCD Enabler. To complete subscription setup, the DCD Enabler validates the subscription with the DCD-Enabled Client Application. It is assumed here that the external subscription portal is associated with the DCD Content Provider. In this scenario the DCD Server is required to obtain device confirmation before setting up the subscription. In the alternative scenario, the subscription portal may be associated with the DCD Service Provider and the DCD Server may receive subscription notification directly from the subscription portal.

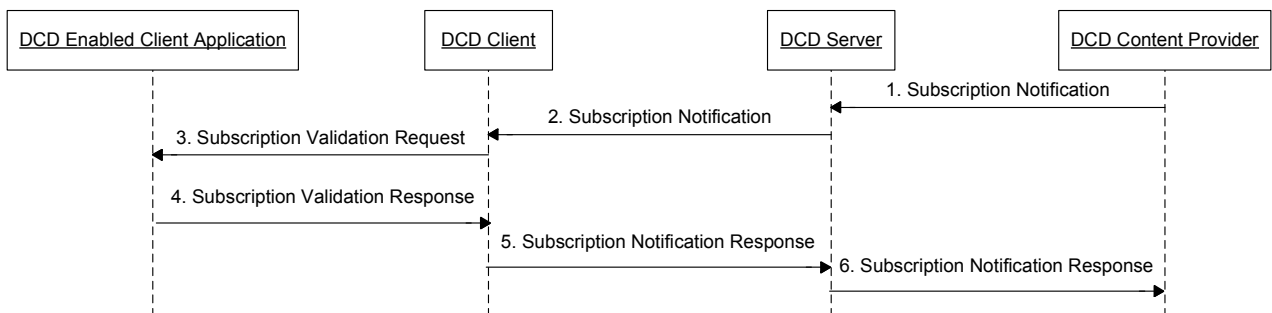


Figure 8: Notification of External Subscription for a DCD Channel

1. DCD Content Provider sends subscription notification to the DCD Server. The subscription notification message contains delivery endpoint information (e.g. device and/or user identity, application identifier, etc.), channel identifier, and subscription identifier, if applicable.
2. DCD Server identifies target device and sends subscription notification to the DCD Client
3. DCD Client requests subscription validation from the target DCD-Enabled Client Application
4. DCD-Enabled Client Application validates the subscription and sends response back to the DCD Client
5. The DCD Client forwards the response to the DCD Server
6. The DCD Server validates application response and, if subscription is confirmed, establishes the subscription. Next, it forwards subscription notification response to the DCD Content Provider.

5.6.5 DCD Content Delivery

5.6.5.1 Pull-based Content Delivery over DCD-1 Interface

This example illustrates pull based content delivery of the DCD Content. The flow describes two scenarios, one with content retrieval from the DCD Content Provider upon receiving a content request and another with content publication at the DCD Server prior to the content request. Both scenarios involve content adaptation at the DCD Server based on the information obtained from personalization sources (e.g. location, presence, etc.)

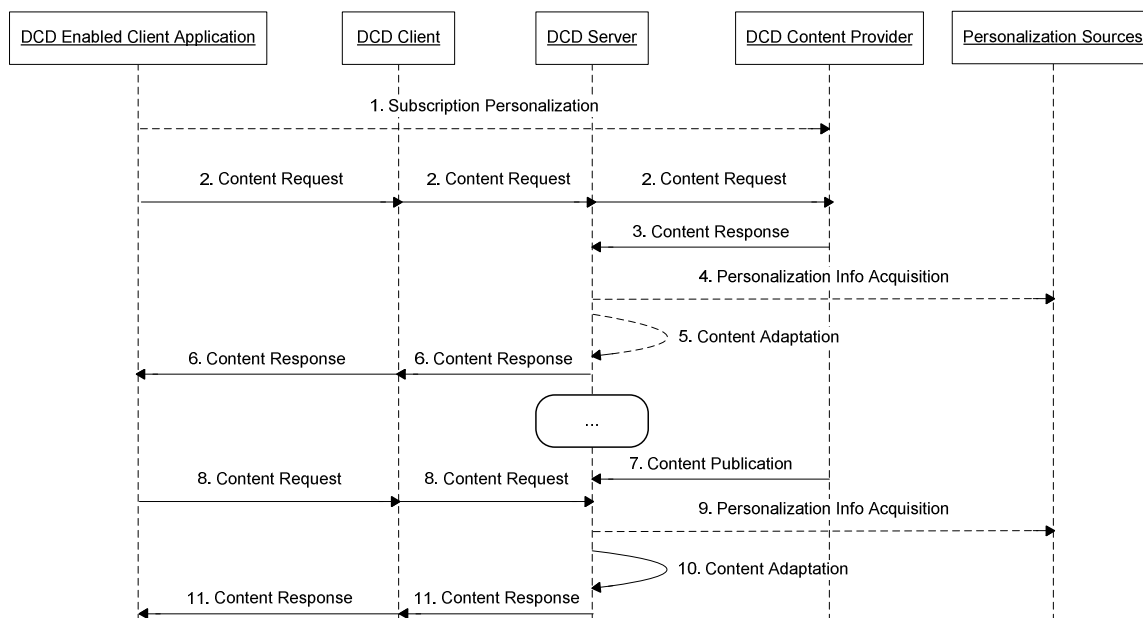


Figure 9: Pull-based Content Delivery over DCD-1 Interface

1. Subscription personalization request is sent from the DCD Enabled Client Application to the DCD Content Provider. Subscription personalization is optional as some DCD Content Providers may not support personalized content. Subscription personalization with the DCD Content Provider typically follows channel subscription with the DCD Enabler (see sections 5.6.3 and 5.6.4).
2. Content request is sent from the DCD Enabled Client Application to the DCD Content Provider. This content request could be a result of user action or a scheduled content query by the DCD Enabled Client Application or by the DCD Client (e.g. based on AP preferences).

3. Content response from the DCD Content Provider to the DCD Server. The response contains the DCD Content or the error information if the content is unavailable.
4. DCD Server acquires personalization information required to perform content adaptation. Such information may include location, presence, terminal profile, etc.
5. DCD Server performs content adaptation based on the acquired personalization information and the Application Profile preferences obtained at registration.
6. Content is returned to the application. Alternatively, the DCD Client may retain the content for further retrieval by the application.
7. DCD Content Provider publishes new content to the DCD Server. This could be either the content matching subscription preferences (as specified at step 1), or some general content of interest to multiple DCD Enabled Client Applications.
8. DCD Enabled Client Application issues a Content Request. The request is terminated at the DCD Server as the content is available locally due to content publication at step 7.
9. Same as step 4
10. Same as step 5
11. Same as step 6

5.6.5.2 Push-based Content Delivery over DCD-2 Interface

This example illustrates push based content delivery of the DCD Content.

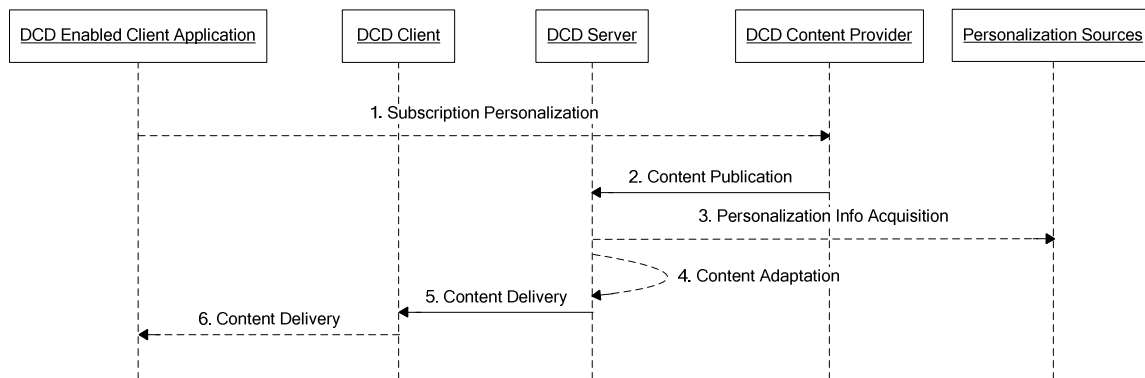


Figure 10: Push-based Content Delivery over DCD-2 Interface

1. Subscription personalization request is sent from the DCD Enabled Client Application to the DCD Content Provider. Subscription personalization is optional as some DCD Content Providers may not support personalized content. Subscription personalization with the DCD Content Provider typically follows channel subscription with the DCD Enabler (see sections 5.6.3 and 5.6.4).
2. DCD Content Provider publishes new content at the DCD Server. This could be either the content matching subscription preferences (as specified at step 1), or some general content of interest to multiple DCD Enabled Client Applications.
3. DCD Server acquires personalization information required to perform content adaptation. Such information may include location, presence, terminal profile, etc.

4. DCD Server performs content adaptation based on the acquired personalization information and the Application Profile preferences obtained at registration.
5. Content is pushed to the DCD Client. If the terminal cannot be reached (e.g. out-of-coverage), the DCD Server retains the content until the terminal is available (with respect to the settings in channel and content metadata). At this time, the server evaluates the content validity based on expiry and replacement information in the content metadata. If the content is still valid, the DCD Server repeats steps 3 and 4 prior to content push.
6. DCD Content is delivered to the DCD Enabled Client Application. Alternatively, the content could be retained by the DCD Client until the application retrieves this content.

5.6.5.3 Pull-based Content Delivery over DCD-1 Interface with Content Availability Notification over DCD-2 Interface

This example shows pull based content delivery of the DCD Content. The DCD Server uses the DCD-2 interface to send a content availability notification to the DCD Client and the application. The flow describes initial content publication and content update scenarios.

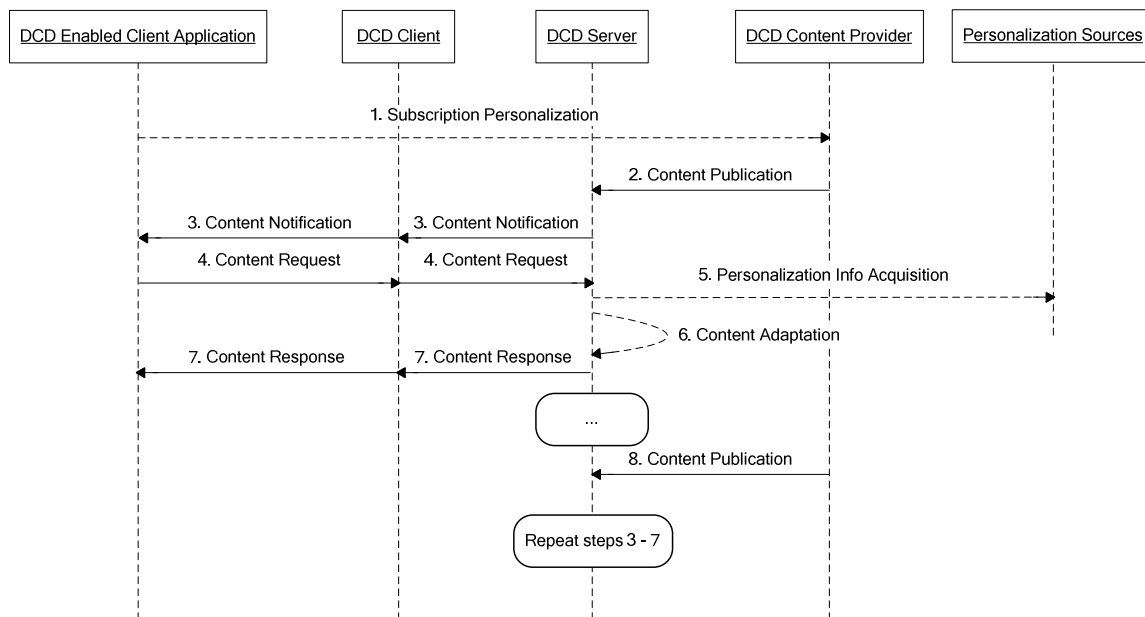


Figure 11: Pull based Content Delivery over DCD-1 Interface with Content Availability Notification over DCD-2 Interface

1. Subscription personalization request from the DCD Enabled Client Application to the DCD Content Provider. Subscription personalization is optional as some DCD Content Providers may not support personalized content. Subscription personalization with the DCD Content Provider typically follows channel subscription with the DCD Enabler (see sections 5.6.3 and 5.6.4).
2. DCD Content Provider publishes new content to the DCD Server. This could be either the content matching subscription preferences (as specified at step 1), or some general content of interest to multiple DCD Enabled Client Applications.
3. DCD Server sends a content availability notification to the DCD Client and the DCD Client relays this notification to the application, where applicable. Alternatively, the DCD Client may retain the notification until it is queried by the application.
4. Content request is sent from the DCD Enabled Client Application to the DCD Server.

5. DCD Server evaluates content validity based on the expiry and replacement information in the content metadata. If the content is still valid, the server acquires personalization information required to perform content adaptation. Such information may include location, presence, terminal profile, etc.
6. DCD Server performs content adaptation based on the acquired personalization information and the Application Profile preferences obtained at registration.
7. Content is returned to the application. Alternatively, the DCD Client may retain the content for further retrieval by the application.
8. DCD Content Provider publishes new or updated content to the DCD Server. The following steps are identical to steps 3-7 above.

5.6.5.4 Suspension and Resumption of Content Delivery Channels

This example shows 2 scenarios for DCD Channel suspension and resumption. The first two scenarios are driven by the DCD Enabled Client application: first directly, requesting the DCD Client to suspend delivery of certain channels, another indirectly through the Application Profile settings. The third scenario is driven by the DCD Content Provider and the DCD Client is notified upon channel suspension and resumption by the content provider.

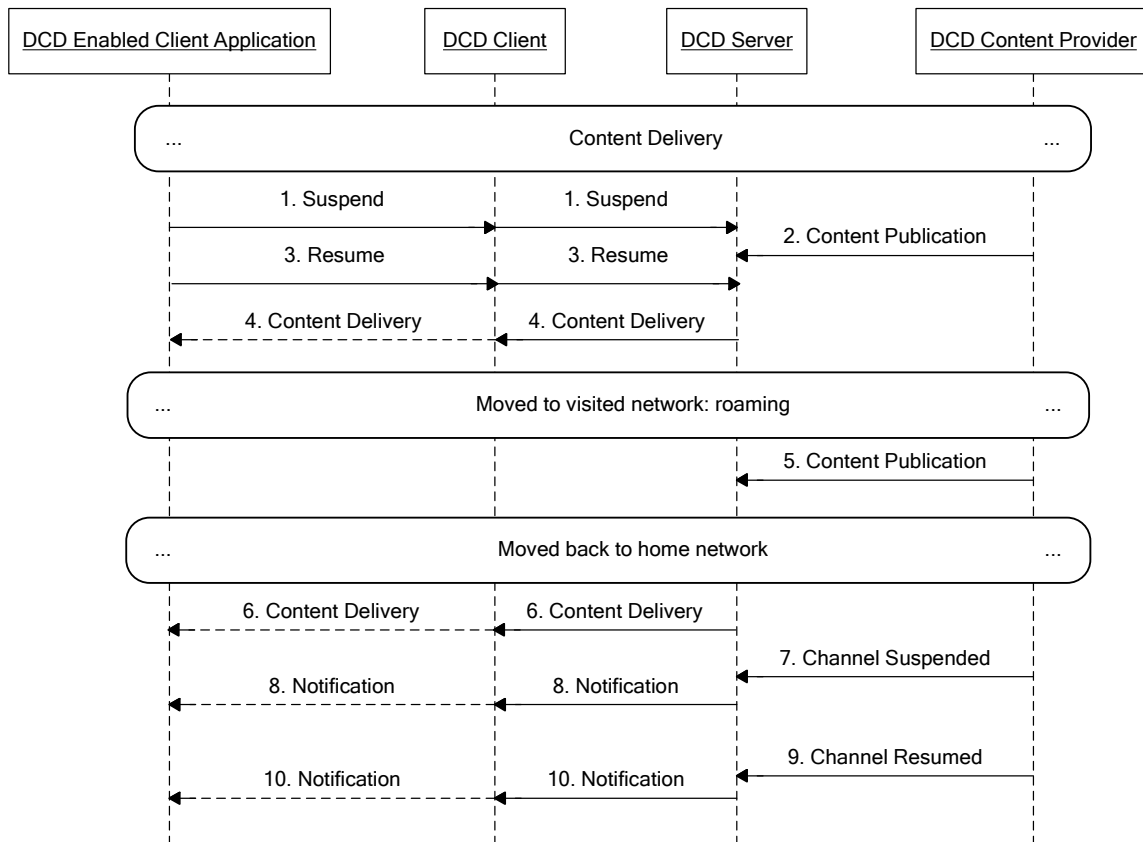


Figure 12: Suspension and Resumption of Content Delivery Channels

1. DCD Enabled Client Application requests the DCD Client to suspend content delivery for a channel or all channels. The DCD Client relays this request to the DCD Server.

2. DCD Content Provider publishes new content to the DCD Server. The DCD Server retains the content until the channel is resumed. Alternatively, the DCD Server may discard the content based on the channel metadata provided by the content provider.
3. DCD Enabled Client Application requests DCD Client to resume content delivery. The DCD Client relays this request to the DCD Server.
4. Content is delivered to the DCD Client and to the DCD Enabled Client Application. Alternatively, the content could be retained by the DCD Client until the application retrieves this content. It is assumed (not shown in the diagram) that at this step the DCD Server may perform content validation (e.g. for expiry) and adaptation.
5. DCD Content Provider publishes new content to the DCD Server. Based on location information, the DCD Server determines that the terminal is roaming. The Application Profile settings prohibit content delivery to the certain channel(s) while roaming. The DCD Server retains the content until the terminal is back to its home network. Alternatively, the DCD Server may discard the content based on channel metadata provided by the content provider. During this period the content delivery is suspended as both the DCD Client and the DCD Server are aware of the suspended state.
6. DCD Server determines that the terminal is back to its home network and delivers the content to the DCD Client and to the DCD Enabled Client Application. It is assumed that at this step the DCD Server may perform content validation (e.g. for expiry) and adaptation.
7. DCD Content Provider notifies the DCD Server about suspension of content delivery for a channel or a group of channels.
8. DCD Server sends a channel suspension notification to the DCD Client and to the DCD Enabled Client Application.
9. DCD Content Provider notifies the DCD Server about resumption of content delivery for a channel or a group of channels.
10. The DCD Server sends channel resumption notification to the DCD Client and to the DCD Enabled Client Application.

5.6.6 Deregistration of the DCD Enabled Client Application

Deregistration process occurs when the DCD Enabled Client Application initiates deregistration request to the DCD Client. Such request could be issued as a result of user action, software removal, etc. Alternatively, the DCD Client may deregister an application without receiving deregistration request (e.g. when discovers that the application is no longer available).

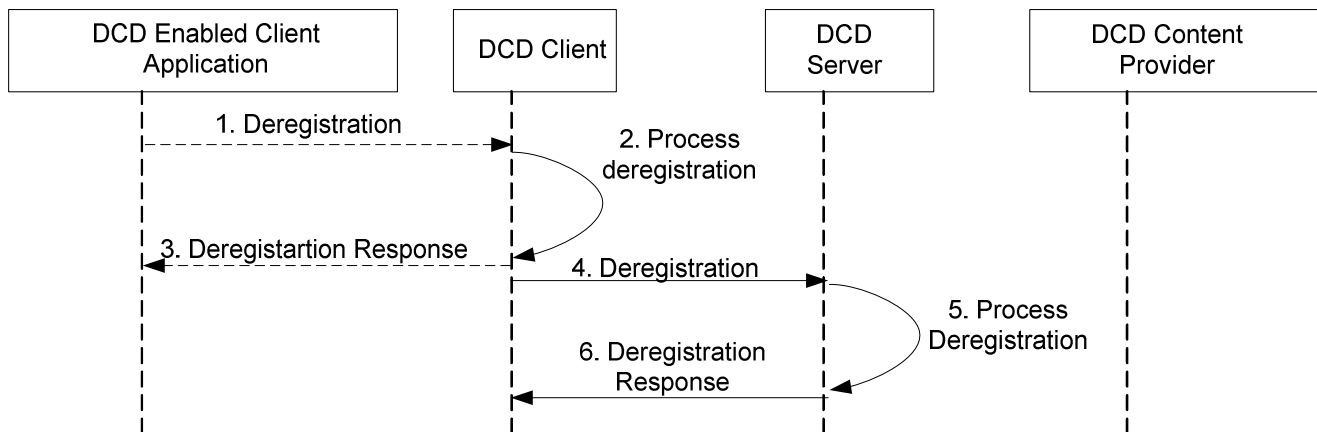


Figure 13: Deregistration of DCD Enabled Client Application

1. DCD Enabled Client Application issues Deregistration request to the DCD Client.
2. DCD Client processes deregistration and may delete the Application Profile context.

3. DCD Client returns Deregistration Response to the DCD Enabled Client Application
4. The DCD Client sends Deregistration request to the DCD Server.
5. DCD Server receives the Deregistration request and processes deregistration (e.g. deletes all application related context for the specific user).
6. DCD Server sends Deregistration Response to the DCD Client

5.6.7 Lifecycle of the DCD Enabled Client Application

This example shows the lifecycle of a DCD Enabled Client Application using point-to-point content delivery over the DCD-1 and DCD-2 interfaces. The following events are shown in the lifecycle diagram:

- New DCD Enabled Client Application joins the DCD Enabler. This action could be triggered by installation of the new application, installation of the DCD Client, a user action, an action of the external application (e.g. content discovery client), a Service Provider action (e.g. through OMA DM), etc.
- The DCD Enabled Client Application is upgraded or modified. As a result of this, the Application Profile needs to be updated (due to a change in application capabilities, in terms of supported channels or channel settings).
- Availability of new content channels as a result of a new channel registration. The analogous event of channel removal results in identical flow and is not shown in the diagram below.
- Terminal switching to roaming results in content delivery suspension, as prescribed in the Application Profile provided at registration.
- Removal of the DCD Enabled Client Application or application deactivation on the terminal.

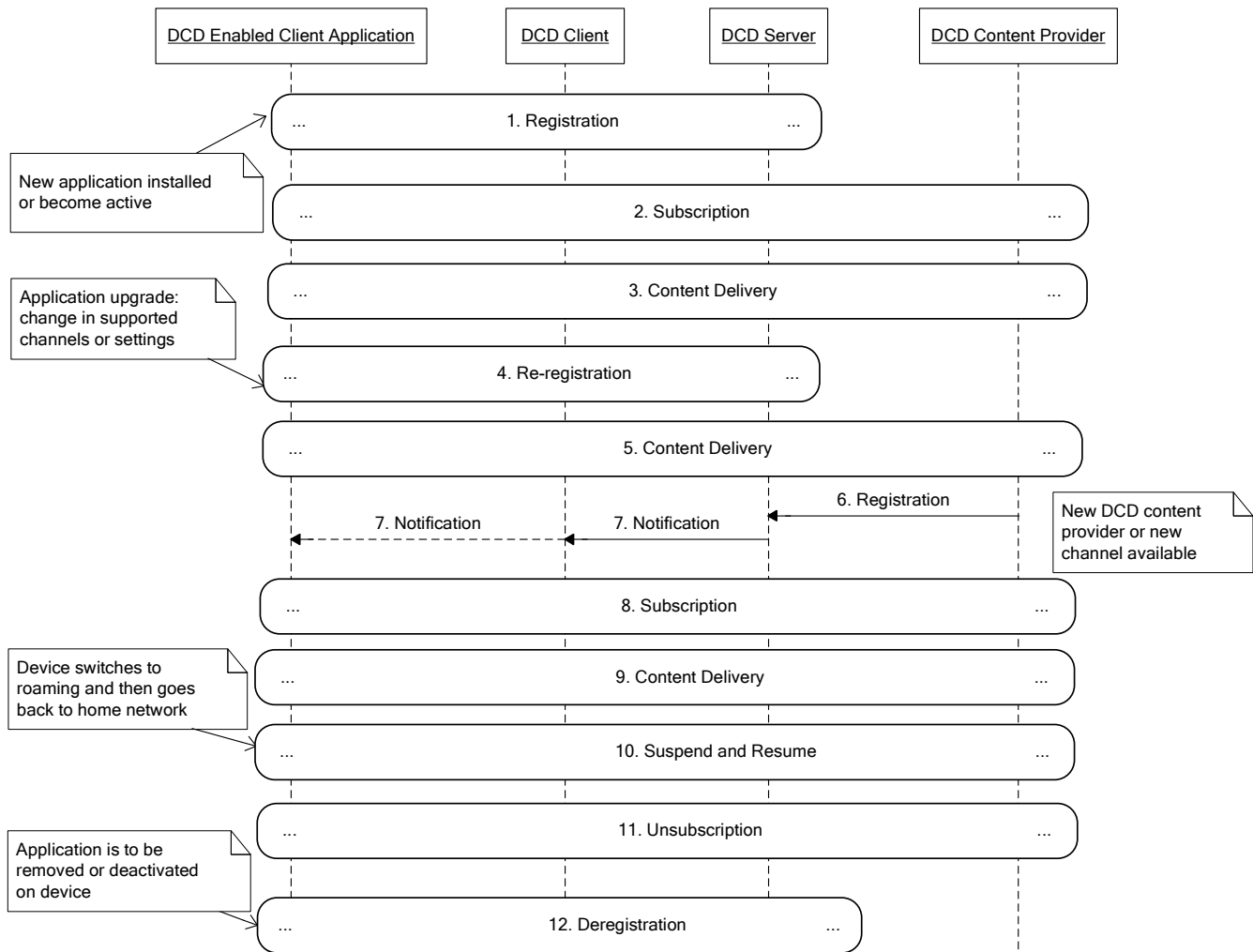


Figure 14: Lifecycle of the DCD Enabled Client Application using P2P content delivery

1. DCD Enabled Client Application registers with the DCD Client and the DCD Client registers the application with the DCD Server. See section 5.6.1 for details. The DCD Client activation with the DCD Server (not shown) is a prerequisite for this step.
2. DCD Enabled Client Application discovers DCD Channels and subscribes with the DCD Enabler. See sections 5.6.3 and 5.6.4 for details.
3. Content is delivered to the DCD Enabled Client Application. The application may need to set up personalized subscription with the DCD Content Provider. See sections 5.6.5.1, 5.6.5.2 and 5.6.5.3 for details.
4. DCD Enabled Client Application reregisters with the DCD Client providing an updated Application Profile. The DCD Client reregisters application with the DCD Server, relaying updated information to the server. The flow is identical to new registration process described in section 5.6.1.
5. Content is delivered to the DCD Enabled Client Application. The application may need to subscribe to new channels with the DCD Content Provider or re-subscribe to previously subscribed channels (e.g. if application ID has changed). See sections 5.6.5.1, 5.6.5.2 and 5.6.5.3 for details.
6. DCD Content Provider registers a new content channel providing the metadata associated with this channel.

7. DCD Server matches channel information with application profile preferences for the registered application and sends new channel availability notifications to applicable DCD Clients. The DCD Client relays this notification to the DCD Enabled Client Application or retains this notification for later retrieval by the application.
8. DCD Enabled Client Application chooses to subscribe for a new channel. See sections 5.6.3 and 5.6.4 for details.
9. Content is delivered to the DCD Enabled Client Application. The application may need to set up personalized subscription for a new channel with the DCD Content Provider. See sections 5.6.5.1, 5.6.5.2 and 5.6.5.3 for details.
10. The terminal moving to a visited network results in content delivery suspension, as prescribed in the Application Profile. Content delivery is resumed when DCD Server determines that the terminal is back to the home network. See section 5.6.5.4 for details.
11. DCD Enabled Client Application unsubscribes some of the DCD Channels.
12. DCD Enabled Client Application deregisters with the DCD Client and the DCD Client sends deregistration message to the DCD Server. All outstanding subscription information and content for this application is deleted by the DCD Enabler. See section 5.6.6 for details.

6. DCD Enabler Protocol Stack

6.1 DCD Enabler Protocol Stack

The DCD Enabler architectural model describes the interactions of DCD Enabler with other Enablers.

Based on the DCD Enabler Architectural model, the DCD Enabler is viewed as a protocol stack with five layers as follows.

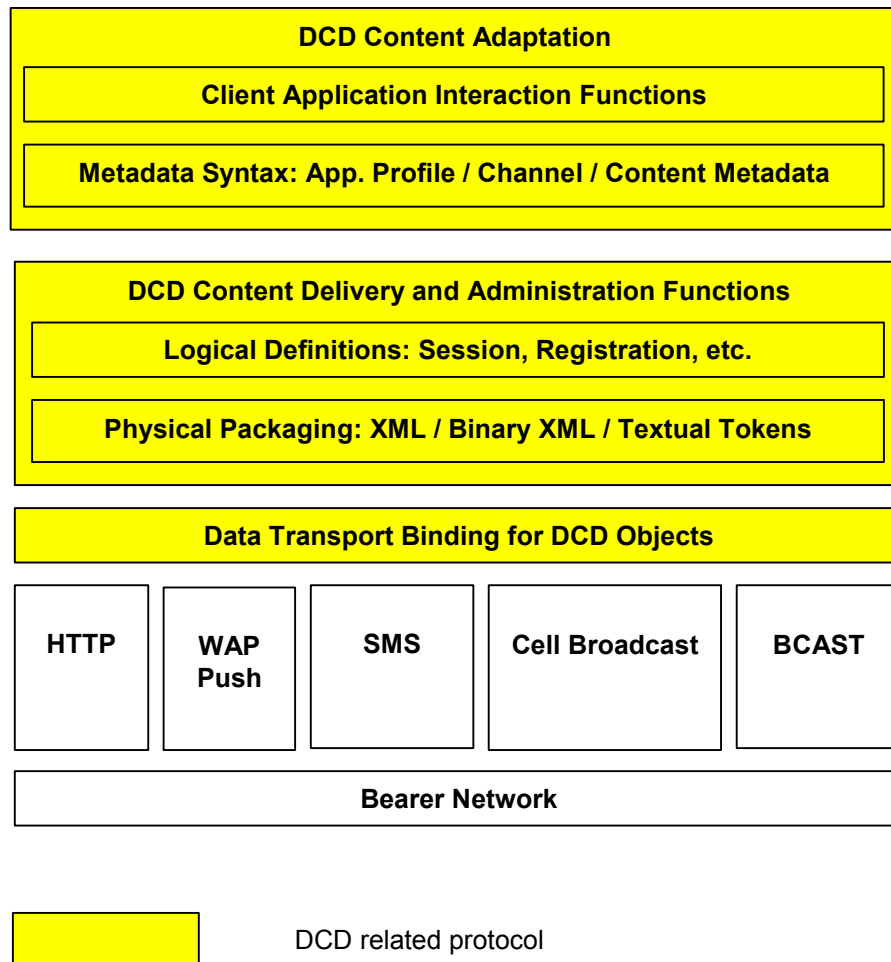


Figure 15: DCD Enabler Protocol Stack

The bottom two layers are the data transport layers over different media, i.e. bearer networks for both point-to-point and point-to-multipoint paradigms. These bottom two layers will reuse the existing bearer protocols for point-to-point and point-to-multipoint data transport, for example HTTP, Cell Broadcast etc.

Above the bottom two data transport layers, the DCD Data Transport Binding defines the interface specifications on how to deliver the DCD Content and related control management messages between the DCD Server and the DCD Client over different data transport layers. The DCD objects encapsulated in different syntax models are transported over data transports. The syntax models, such as XML, binary XML object and textual token objects, are defined as part of DCD Content Delivery and Administration Functions.

The DCD Content Delivery and Administration Functions define two aspects of the key DCD Enabler functions. One is the logical definition of the DCD Enabler functions including the control management message types and informational elements. Thus the DCD Client and DCD Server can interoperate in terms of control management functions. The other is the

physical packaging of the logical functions, e.g. the standard payload structure of the DCD Content and related control management messages. There may be several types of payload structure depending on the transport bearer limitations. For example, it may be XML for HTTP, or an SMS token table for Cell Broadcast, etc. However, the application-specific data is opaque to the content delivery layer, and the definitions and interpretation rules are represented in Application Profile.

At the top, there is the Application Profile (AP) (including Channel Metadata) that allows the DCD Client to handle the DCD Content for delivery to a particular DCD-Enabled Client Application. The Client Application Interaction Functions use Application Profile settings to process the DCD Content. At this layer, the channel and / or user-specific functions can be applied, e.g. through integration with other Enablers not directly related to content delivery.

Appendix A. Other OMA Enabler Entity Use by DCD (Informative)

This appendix describes DCD enabler entity direct use of, or indirect support by, other OMA enabler entities. The entities described in this appendix are not directly dependencies of the DCD Enabler, but deployment options for the fulfillment of specific DCD requirements. For example, to personalize/customized DCD operation, the DCD Server can retrieve personalization/customization-affecting information from other enablers.

While OMA seeks to promote the use of OMA enablers for such functions, there are other options which DCD deployments can include. For example non-OMA location/presence facilities can provide personalization/customization-affecting information, and DCD entities may obtain this information through other unspecified means. However, where a deployment does call for use of OMA Enablers for these functions, it is expected that DCD entity implementations must conform to the OMA requirements for interfacing with those enablers. This section thus describes the OMA enabler entities and their interfaces for use by DCD, if applicable for a specific deployment.

A.1 OMA Device Management Enabler

A.1.1 Device Provisioning and Management Client

The Device Provisioning and Management Client is an external entity providing the provisioning functions to the DCD Client. It is able to:

- Receive the initial parameters needed for DCD Service sent by service provider by using mechanisms specified in [OMA-CP-ARCH] or [OMA-DM-AD].
- Retrieve and update the parameters needed for DCD applications sent by a Service Provider by using [OMA-DM-AD].

The detailed description of the functionalities is given in [OMA-CP-ARCH], [OMA-DM-AD], [OMA-FUMO-AD] and [OMA-SCOMO-AD].

Editor's note: the specific interfaces to be used and other dependencies are TBD pending further discussion with the DM group.

A.1.2 Device Provisioning and Management Server

The Device Provisioning and Management Server may perform the following functions that are needed in support of DCD applications:

- Initialization and updating of all the configuration parameters necessary for the DCD Client.
- Software update for terminals allowing OTA application upgrade.

The detailed description of the functionalities is given in [OMA-CP-ARCH], [OMA-DM-AD], [OMA-FUMO-AD] and [OMA-SCOMO-AD].

Editor's note: the specific interfaces to be used and other dependencies are TBD pending further discussion with the DM group.

Charging Entity

A.2 OMA Charging Enabler

A.2.1 Charging Entity

This is an external entity, which may reside in the Service Provider's domain. This entity takes various roles, which network operators and / or service providers need to perform charging activities.

This entity is specified in [OMA-MCC-AD].

Editor's note: the specific interfaces to be used are TBD pending further discussion with the MCC group.

Presence Server

A.3 OMA Presence Enabler

A.3.1 Presence Server

The Presence Server provides presence information of a user, which may be used for content personalization.

This entity is specified in [OMA-PRS-IMPS-AD] and [OMA-PRS-SIMPLE-AD].

For access to presence information, the DCD Server acts as a Watcher and interfaces to:

- the SIP/IP Core network via the PRS-2 reference point, as defined in [OMA-PRS-SIMPLE]
- the Content Server via the PRS-14 reference point, as defined in [OMA-PRS-SIMPLE]

The DCD Server uses the methods defined by [OMA-PRS-SIMPLE] to acquire user presence information.

The conditions under which the Presence Service is invoked and the methods to be used in any specific DCD supported services are application-specific, and thus not defined or limited by the DCD Enabler.

Location Server

A.4 OMA Location Enabler

A.4.1 Location Server

The Location Server is an entity that provides location information of a user utilizing DCD supported services.

This entity is specified in [OMA-MLS-AD].

For access to location information, the DCD Server acts as a MLS Client and interfaces to:

- the Requesting Location Server via the Le reference point, using the Mobile Location Protocol (MLP) V3.2 as defined in [OMA-MLP3.2].

The DCD Server uses the HTTP-based methods defined by MLP V3.2 to acquire user location information.

The conditions under which the Location Server is invoked and the methods to be used in any specific DCD supported services are application-specific, and thus not defined or limited by the DCD Enabler.

Categorization-Based Content Screening Enabler Content Screening Entity

A.5 OMA Categorization-Based Content Screening Enabler

A.5.1 Content Screening Entity

The Content Screening Entity of the Categorization-Based Content Screening Enabler (CBCS) provides content screening functionality.

This entity is specified in [OMA-CBCS-AD].

The DCD Server and DCD Client interact with the CBCS Content Screening Component via the CBCS-1 interface defined in [OMA-CBCS-AD].

The conditions under which the CBCS Content Screening Entity is invoked and the methods to be used in any specific DCD supported services are application-specific, and thus not defined or limited by the DCD Enabler.

As the CBCS Enabler can alter the DCD content workflow, the DCD Enabler defines the error codes associated with CBCS blocking the content during delivery.

Client-Side Content Screening Framework Service Scan Engine

A.6 OMA Client-Side Content Screening Framework Enabler

A.6.1 Scan Engine

The Scan Engine entity of the Client-Side Content Screening Framework (CSCF) provides content screening functionality in terminals.

The CSCF Scan Engine is specified in [OMA-CSCF-AD].

The DCD Client interacts with the Scan Engine via the CSF-1 interface as specified in [OMA-CSCF-TS].

The conditions under which the CSCF Scan Engine is invoked and the methods to be used in any specific DCD supported services are application-specific, and thus not defined or limited by the DCD Enabler.

As the CSCF Enabler can alter the DCD content workflow, the DCD Enabler defines the error codes provided by the DCD Client associated with CSCF blocking the content during delivery.

User Agent Profile Storage Server

A.7 OMA User Agent Profile Enabler

A.7.1 User Agent Profile Storage Server

The User Agent Profile Storage Server is an entity of the UAProf framework that stores terminal's profile resources. The User Agent Profile Storage Server [OMA-UAPROF] will provide information for content adaptation.

This entity is specified in [OMA-UAPROF].

DRM Enabler Entities

A.8 OMA DRM Enabler

The DRM Enabler is used in order to ensure controlled consumption of digital media objects by providing the ability, for example, to manage previews of DRM Content, to enable superdistribution of protected contents, and to enable transfer of content between DRM Agents. DRM Enabler entities are defined in [OMA-DRM].

In the DRM framework, DCD Server acts as content issuer that is an entity that delivers DRM Content. [OMA-DRM] defines the format of protected content delivered to DRM Agents, and the way protected content can be transported from a content issuer to a DRM Agent using different transport mechanisms. The content issuer (i.e. DCD Server) may do the actual packaging of protected content itself, or it may receive pre-packaged protected content from DCD Content Provider.

While DCD Server assumes role of content issuer in the DRM framework, it does not directly interact with any entities of DRM Enabler, but permits the DCD Content to be protected by using the DRM Content Format specification. DCD Content payload is opaque to DCD Enabler. DCD Content payload can be either unprotected or protected.

On the terminal side, the DRM Agent is responsible for enforcing permissions and constraints associated with protected content, controlling access to protected content, etc. As described earlier, DCD-Enabled Client Application consumes content that is transparently made available by the DCD Enabler. DCD-Enabled Client Application may interact with DRM Agent when DCD Content payload is protected. Hence, DCD Client does not directly interact with any entities of DRM Enabler either.

Appendix B. DCD over BCAST Architecture

B.1 Architectural Diagram

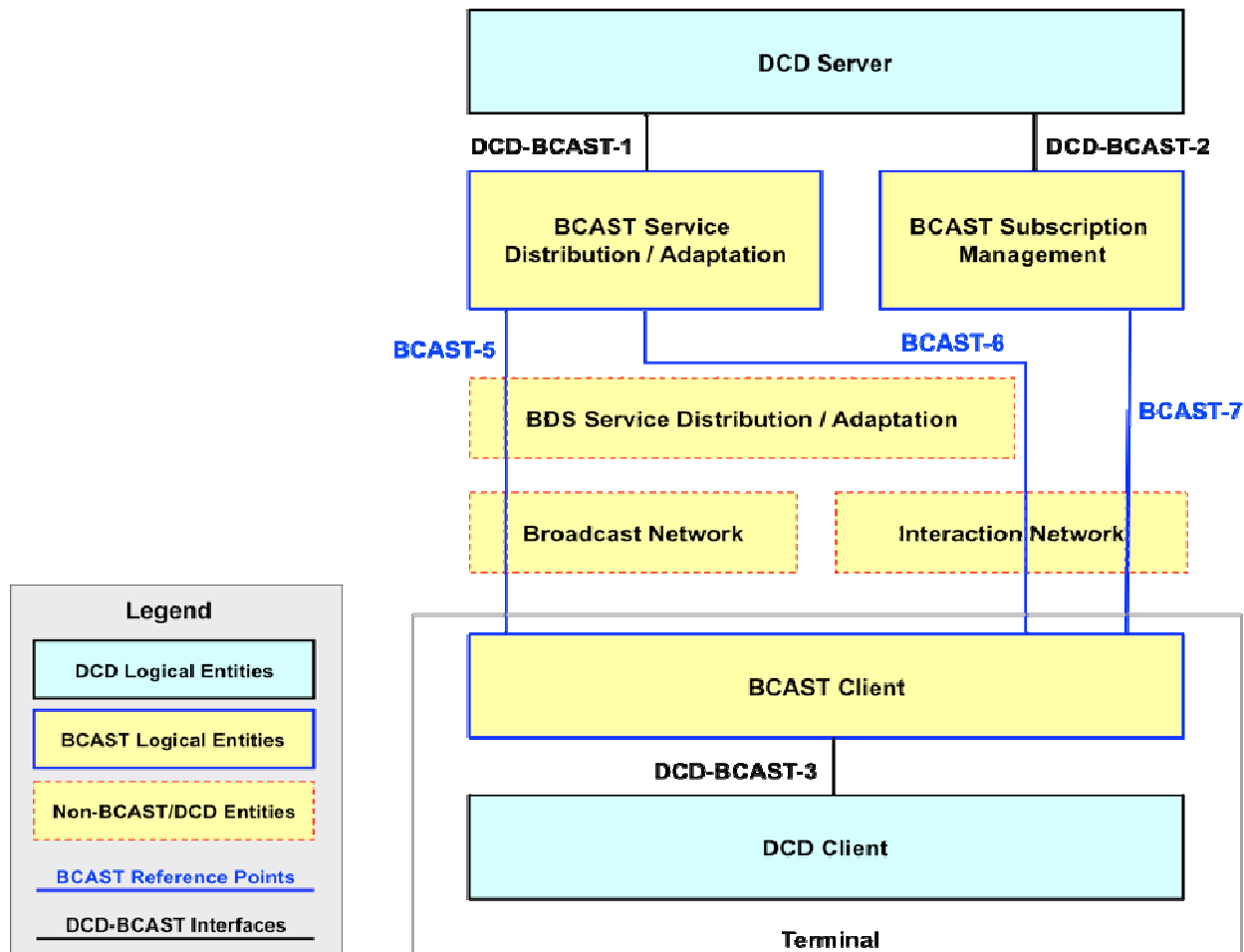


Figure 16: DCD over BCAST Functional Architecture Diagram

Figure 1 provides an overview of the DCD over BCAST functional architecture. This architecture is used for push delivery of DCD content over BCAST as part of the DCD-2 interface and push delivery of administrative messaging over BCAST as part of the broadcast instantiation of the DCD-3 interface.

The DCD over BCAST architecture includes entities and interfaces from the OMA BCAST enabler. The architecture also includes a set of BCAST adaptation specific DCD interfaces (DCD-BCAST) which enable interaction between the generic DCD enabler entities and corresponding BCAST enabler entities.

A description of each of the interfaces and reference points shown in Figure 1 and their role in the DCD over BCAST architecture is provided in the sections that follow.

B.2 BCAST Reference Points

The DCD over BCAST adaptation utilizes interfaces from the BCAST-5, BCAST-6, and BCAST-7 reference points described in [OMA-BCAST-AD] to provision broadcast sessions, enable service protection, and deliver DCD content over BCAST. The BCAST reference points relevant to the DCD over BCAST adaptation are described below.

B.2.1 BCAST-5 Reference Point

The DCD over BCAST adaptation utilizes the following interfaces that are defined as part of the BCAST-5 reference point described in [OMA-BCAST-AD]:

Interface	Usage
SG-5	The SG-5 interface may be used for the delivery of BCAST Service Guide containing BCAST file delivery session access parameters for DCD channels over the Broadcast Distribution System.
FD-5	The FD-5 interface is used for the delivery of BCAST files or bundles of files containing DCD content and associated content metadata over the Broadcast Distribution System.
SP-5-1	The SP-5-1 interface is used for the delivery of service protected BCAST files or bundles of files containing DCD content and associated metadata which requires service protection.
SP-5-2	The SP-5-2 interface is used for the delivery of BCAST STKM over the broadcast channel to the BCAST SP-C in terminal or smartcard when BCAST service protection is desired.

B.2.2 BCAST-6 Reference Point

The DCD over BCAST adaptation utilizes the following interfaces that are defined as part of the BCAST-6 reference point described in [OMA-BCAST-AD]:

Interface	Usage
SG-6	The SG-6 interface may be used for the delivery of BCAST Service Guide containing BCAST file delivery session access parameters for DCD channels over the Interaction Channel.
FD-6	The FD-6 interface is used for point-to-point delivery of BCAST files or bundles of files containing DCD content and associated metadata over the Interaction Channel.
SP-6	The SP-6 interface is used for delivery of BCAST STKM over the interaction channel when BCAST service protection is desired.

B.2.3 BCAST-7 Reference Point

The DCD over BCAST adaptation utilizes the following interfaces that are defined as part of the BCAST-7 reference point described in [OMA-BCAST-AD]:

Interface	Usage
SP-7	The SP-7 interface may be used for the signalling exchange for registration and delivery of LTKM over the interaction channel for BCAST file delivery sessions associated with DCD channels.
SPR-7	The SPR-7 interface may be used for delivery of messages used for subscription to BCAST file delivery sessions associated with DCD channels.

B.3 DCD over BCAST Interfaces

The DCD Server must interact with the BCAST Service Distribution / Adaptation function and the BCAST Subscription Management function to deliver DCD content and administrative messaging over BCAST. This interaction is accomplished via the DCD-BCAST-1 and DCD-BCAST-2 interfaces respectively.

The DCD Client must interact with the BCAST Client to receive DCD content and administrative messaging over BCAST. This interaction is accomplished via the DCD-BCAST-3 interface.

B.3.1 DCD-BCAST-1 Interface

The DCD-BCAST-1 interface allows the DCD Server to interact with the BCAST Service Distribution / Adaptation function to provision broadcast sessions and to insert DCD content into the BCAST enabler. This interface is expected to allow the DCD Server to perform the following functions:

- Establish a BCAST file delivery session
- Submit DCD content to be delivered on a specific BCAST file delivery session
- Remove DCD content from a specific BCAST file delivery session
- Terminate a BCAST file delivery session
- Obtain or specify access parameters or a service guide fragment reference for a BCAST file delivery session

B.3.2 DCD-BCAST-2 Interface

The DCD-BCAST-2 interface allows the DCD Server to interact with the BCAST Subscription Management function to provision subscription information for DCD channels distributed over BCAST. This interface is expected to allow the DCD Server to perform the following functions:

- Provision subscription and authorization related information required by the BCAST subscription management function

B.3.3 DCD-BCAST-3 Interface

The DCD-BCAST-3 interface allows the DCD Client to interact with the BCAST Client to initiate content reception and to receive DCD content delivered over BCAST. This interface is expected to allow the DCD Client to perform the following functions:

- Initiate content reception on a specific BCAST file delivery session by providing BCAST access parameters obtained through DCD metadata
- Obtain content delivered on BCAST file delivery sessions for which content reception has been initiated
- Terminate content reception on a specific BCAST file delivery session

B.4 DCD over BCAST Metadata

B.4.1 Inclusion of BCAST Access Information in DCD Metadata

Access information for the BCAST file delivery session associated with each DCD channel or interface delivered over BCAST must be available to the DCD Client in order for the DCD Client to initiate content reception over BCAST. This information is provided to the DCD Client through DCD metadata. There are several mechanisms through which the DCD Client can obtain this information:

- Access information for the DCD-3 interface and for default or pre-subscribed channels delivered over BCAST may be pre-provisioned as part of the Application Profile for an application

- Access information for specific DCD channels delivered over BCAST is provided by the DCD Server as part of the Channel Metadata. This information is available both through channel discovery and in response to a successful channel subscription

B.4.2 Supported BCAST Access Information Formats

BCAST access information may be included in DCD metadata in several different formats. The following formats are supported for inclusion of BCAST access information in DCD metadata:

- URI referencing a BCAST Service Guide ‘Service’ fragment as defined in [OMA-TS-BCAST]
- One or more complete, well-formed BCAST Service Guide ‘Access’ fragments as defined in [OMA-TS-BCAST]
- One or more complete, well-formed SDP session descriptions as defined in [RFC 4566]

Appendix C. Change History

(Informative)

C.1 Approved Version History

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

C.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-AD-DCD-V1_0	26 Oct 2005	All	Initial draft template
	06 Mar 2006	All	Updated according to minutes 0091 and contribution 0074R02. Editorial changes to references.
	16 Mar 2006	All	Updated according to minutes 0109 and contribution 0096R02 and 0098R01.
	05 Apr 2006	All	Updated according to minutes 0154 and 0167, and contributions 0100R01, 0123, 0132R01 and 0141R01.
	27 Apr 2006	All	Updated according to minutes 0171 and 0175, and contribution 0122R02 and 0169R01.
	05 May 2006	All	Updated according to minutes 0167, and contribution 0140R01.
	11 May 2006	All	Updated according to minutes 0197 and 0198, and contributions 0136R01, 0138R01, 0182R01, 0183R01, 0184R01, 0185R01, 0187, 0188, 0190R01, 0194R01 and 0196R01
	26 May 2006	All	Updated according to minutes 0223, and contribution 0218R01
	02 Jun 2006	All	Updated according to minutes 0231 and contribution 0225 and 0230.
	16 Jun 2006	All	Updated according to minutes 0278 and contributions 0195R03, 0214R01, 0226R01, 0241, 0243R01, 0246R01, 0261R01 and 0262R01.
	12 Jul 2006	All	Updated according to minutes 0288 and BAC-CD-2006-0027, and contribution 0287 and BAC-CD-2006-0028.
	21 Jul 2006	All	Updated according to minutes 0031, Bitfone's comments, and contribution 0256, 0269R02 and 0289R02 and 0032
	26 Jul 2006	All	Editorial update according to the discussion and minutes 0036.
	02 Aug 2006	All	Update according to discussion minutes 0042 and contribution 0041.
	15 Sep 2006	All	Updated according to Beijing discussion minutes 0075R02 and contribution 0045R01
	21 Sep 2006	All	Updated according to contribution 0097R01 (minutes 0108).
	14 Oct 2006	All	Updated according to contribution 0072R03, 0083R01, 0087R01, 0090R01, 0092R01, 0093R01, 0095R01, 0096, 0098R01, 0102 and editorial changes (minutes 0103, 0107, 0108, 0113, 0115, 0120).
	06 Nov 2006	All	Updated according to contributions 0110R01, 0111R01, 0122, 0123R01, 0124R01, 0129R02, 0130R03, 0131, 0133R01, 0134R01, 0135, 0137, 0139, 0140R02, 0141R01, 0143R01 and 0144R01 (minutes 0149R01).
	10 Nov 2006	All	Editorial changes (minutes 0149R01 and 0154R01)
	04 Dec 2006	All	Editorial changes (minutes 0154R01 and 0160)
	08 Dec 2006	All	Updated according to contribution 0164R01, 0171, 0177R03 and 0178R01 (minutes 0173 and 0181).
	18 Dec 2006	All	Updated according to contribution 0179, 0183R01, 0188R02, 0192, 0193, 0194R01 (minutes 0195R01)
	04 Jan 2007	All	Updated according to 2006-0174R01 and 2006-0175R01
	01 Feb 2007	All	Updated according to 0018R01, 0019R01, 0027R01 and 0033R01, and editorial changes (minutes 0032).
	14 Feb 2007	All	Updated according to 0047 and 0052 (minutes 0054)
	07 Mar 2007	All	Updated according to BAC CD-2007-A001
	30 Apr 2007	All	Updated according to 0094, 0103R01 and 0104R01 (minutes 0106)
10 May 2007	All	(1) editorial changes of diagram Figure 1 and 2. (2) Updated according to 0122 (minutes 0127)	
22 May 2007	All	Updated according to 0128 (minutes 0145) and editorial changes of cross references of sections	

Document Identifier	Date	Sections	Description
	17 Jul 2007	All	Updated according to CD-DCD-2007-0005R01 and 0016R01 (minutes CD-DCD-2007-0041R01), and updated Figure 9, 10 and 11 according to CD-2007-0128.
	08 Aug 2007	All	Updated according to 0038R01 (minutes 0059)
	11 Sep 2007	All	Updated according to 0085 (minutes 0100)
	19 Nov 2007	5.5.2.1.2	Updated according to 0141R01 (minutes 0154)
	1 May 2008	Appendix A, Appendix B	Updated according to CD-DCD-2008-0089R01 Corrected titles for Appendix A.1 and A.1.2
	08 Sept 2008	All	Updated according to resolutions of all "CLOSED" CONR comments in OMA-CONRR-DCD-V1_0-20080903-I.
Candidate Version OMA-AD-DCD-V1_0	23 Dec 2008	All	Status changed to Candidate by TP OMA-TP-2008-0493- INP_DCD_V1_0_ERP_for_Candidate_Approval