Device Management Architecture

Candidate Version 1.3 – 02 June 2009

Open Mobile Alliance

OMA-AD-DM-V1_3-20090602-C

Contents

Figures

1. Scope (Informative)

The scope of the Device Manegement architecture document is to define the architecture for the Device Management v1.3 enabler. This document fulfils the functional capabilities and information flows needed to support this enabler as described in the Device Management requirements document [DM-RD].

2. References

2.1 Normative References

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

[DM-RD] "Device Management Requirements", Open Mobile Alliance™, OMA-RD-DM-V1_3, URL:http://www.openmobilealliance.org/

[RFC4346] "The Transport Layer Security (TLS) Protocol", T.Dierks, E. Resorla, April 2006, URL:http://www.ietf.org/rfc/rfc4346.txt

[SIPPush] "Push using SIP". Open Mobile Alliance™. OMA-TS-SIP_Push-V1_0.

URL:http://www.openmobilealliance.org

[COMMON12] "Enabler Release Definition for SyncML Common Specifications, version 1.2". Open Mobile Alliance™. OMA-ERELD-SyncML-Common-V1_2. URL:http://www.openmobilealliance.org

[COMMON13] "Enabler Release Definition for SyncML Common Specifications, version 1.3". Open Mobile Alliance™. OMA-ERELD-SyncML-Common-V1_3. URL:http://www.openmobilealliance.org

2.2 Informative References

[ARCH-PRINC] "OMA Architecture Principles", OMA-ArchitecturePrinciples-V1_2, URL:http://www.openmobilealliance.org/

[OMA-DICT] "Dictionary for OMA Specifications", Version 2.7, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_7, URL:http://www.openmobilealliance.org/

[DM-DICT] "Device Management Dictionary", Version 1.0, Open Mobile Alliance™, OMA-SUP-DM-DM_Dictionary-V1_0, URL:http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

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

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

3.2 Definitions

Device	See [OMA-DICT].
Device Management Client	See [DM-DICT].
Device Management Server	See [DM-DICT].
Interface	See [OMA-DICT].
Management Object	See [OMA-DICT].

3.3 Abbreviations

DM	Device Management
OMA	Open Mobile Alliance
MO	Management Object

4. Introduction (Informative)

Device Management refers to the management of Device configuration and other managed objects of Devices from the point of view of the Management Authorities. Device Management includes, but is not restricted to setting initial configuration information in Devices, subsequent updates of persistent information in Devices, retrieval of management information from Devices, execute primitives on Devices, and processing events and alarms generated by Devices.

Device management allows wireless operators, service providers or corporate information management departments to carry out the procedures of configuring devices on behalf of the end user (customer).

4.1 Version 1.2

Device management is the generic term used for technology that allows third parties to carry out the difficult procedures of configuring devices on behalf of the end user (customer). Third parties would typically be operators, service providers or corporate information management departments.

Through device management, an external party can remotely set parameters, conduct troubleshooting servicing of terminals, install or upgrade software. In broad terms, device management consists of three parts:

- Protocol and mechanism: The protocol used between a management server and a device
- Data model: The data made available for remote manipulation, for example browser and mail settings
- Policy: The policy decides who can manipulate a particular parameter, or update a particular object in the device

The specifications in the Device Management enabler v1.3 address the first part of device management above, the protocol and mechanism. More particularly, this enabler release addresses the management of devices by specifying a protocol and management □echanism that may be exposed by an OMA DM client and targeted by an OMA DM server.

The architecture of the Device Management enabler anticipates the needs of the market actors to differentiate their products through vendor-specific extensions while providing a core parameter set that can be relied upon in all terminals exposing this standardized interface.

The design of the architecture follows the OMA architecture principle [ARCH-PRINC] of Network Technology Independence by separating the bearer-neutral requirements from bearer-specific bindings. The described architecture also anticipates additional bearer and proxy types, as any are identified, without requiring a respecification of previously released documents. This preserves vendor and customer investment while supporting the scaling required by future innovations.

There are three parts to the object schema that provide break-points between more general and more specific parameters:

- A top level management object which is bearer-neutral;
- A set of bearer-specific parameters;
- Sub-tree(s) for exposing vendor-specific parameters.

By composing the management objects in this way, it becomes possible for a device management authority to:

- Target generic requirements that span all implementations;
- Focus on bearer-specific idiosyncracies of a given networking environment;
- Activate terminal-specific behaviour by adjusting vendor-specific parameters.

In a wireless environment, the crucial element for device management protocol is the need to efficiently and effectively address the characteristics of devices including low bandwidth and high latency and to provide for support of these management operations remotely, over-the-air.

4.2 Version 1.3

OMA DM Version 1.3 makes no change to the architecture from OMA DM 1.2, but does introduce new notification and transport protocols.

5. Architectural Model

5.1 Dependencies

DM 1.3 will have the same dependencies as the DM 1.2 enabler [DM12] with the following additions/changes:

DM 1.3 has a dependency upon SyncML Common 1.3 [COMMON13]. Note that DM 1.2 depended upon SyncML Common 1.2 [COMMON12].

Additionally, DM 1.3 optionally depends on:

- OMA SIP Push 1.0 enabler [SIPPUSH].
- TLS 1.1 [RFC4346].

5.2 Architectural Diagram

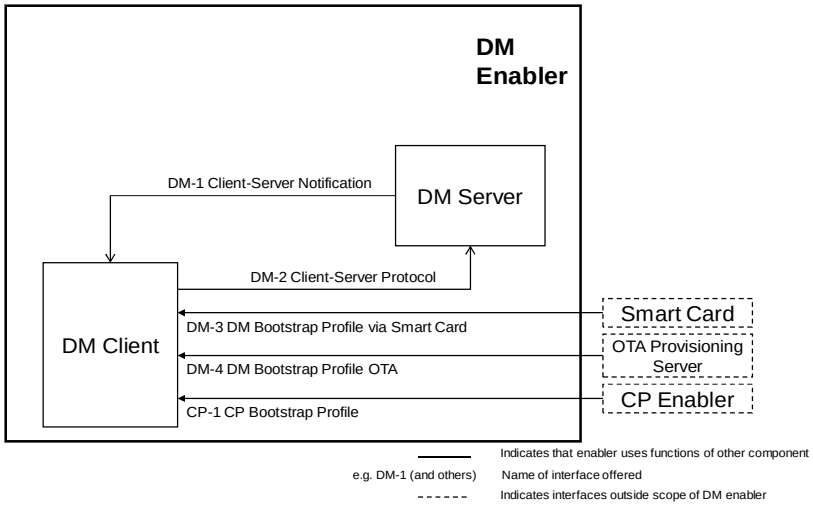

Figure 1: Device Management Architecture using interfaces

5.3 Functional Components and Interfaces/reference points definition

5.3.1 Protocol Endpoints

5.3.1.1 DM Client

The DM Client is the abstract software component that conforms to the requirements for DM Clients specified in the OMA Device Management Enabler.

5.3.1.2 DM Server

The DM Server is the abstract software component that conforms to the requirements for DM Servers specified in the OMA Device Management Enabler.

5.3.2 Interfaces

5.3.2.1 DM-1 Client-Server Notification

This provides an interface over which Servers may send device management notification to Clients. This is an interface that is bearer neutral and can operate over many protocols such as WAP Push and SIP Push.

5.3.2.2 DM-2 Device Management Client-Server Protocol

This provides an interface over which Servers may send device management commands to Clients and Clients may return status and alerts to Servers. This is an interface that is bearer neutral and offers many standardized bindings including HTTP and HTTPS. This interface MAY be exposed over an airlink-based data bearer protocol (e.g. GPRS) to provide over-the-air device management capability.

5.3.2.3 DM-3 DM Bootstrap Profile via Smart Card

The {DM Client} may be initially provisioned via a file on a Smart Card. This file contains a series of DM Commands to set or replace configuration settings in the {DM Client}. This is a one-way interface with no feedback from the DM Client. The only expected result is the {DM Client} connecting to the {DM Server} at the next practical opportunity.

5.3.2.4 DM-4 DM Bootstrap Profile OTA

The {DM Client} may be initially provisioned via a file sent by some push protocol This file contains a series of DM Commands to set or replace configuration settings in the {DM Client}. This is a one-way interface with no feedback from the DM Client. The only expected result is the {DM Client} connecting to the {DM Server} at the next practical opportunity.

5.3.2.5 CP-1 CP Bootstrap Profile

The {DM Client} may be initially provisioned via the {CP enabler}. This is a one-way interface with no feedback from the DM Client. The only expected result is the {DM Client} connecting to the {DM Server} at the next practical opportunity.

Appendix A. Change History (Informative)

A.1 Approved Version History

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

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

Document Identifier	Date	Sections	Description
Draft Versions OMA-AD-DM-V1_3	10 Jul 2003	All	New baseline
	22 Sep 2008	4.1, 5.2, 5.3,	Applied CRs OMA-DM-2008-0126R04-CR_AD_diagram and OMA-DM-2008-0136-CR_AD_Functions.
	22 Oct 2008	5	Applied CRs OMA-DM-2008-0146R01-CR_AD_Flows and OMA_DM-2008-0147R01-CR_AD_Security
	12 Jan 2009	2.1, 4.2, 5.1 and B	Applied CR OMA-DM-DM13-2008-0015-CR_AD_Cleanup.
	29 Jan 2009	2.1 and 5.1	Applied CR OMA-DM-2008-0173R03-CR_Dependencies.
	22 Apr 2009	All	Applied CR OMA-DM-DM13-2009-0015R02-CR_AD_Comment_Resolutions
	11 May 2009	All	Accepted change bars from uploaded version dated 22 April. Editorial Fixes to cover page i.e. date.
Candidate Versions: OMA-AD-DM-V1_3	02 June 2009	N/A	Status changed to Candidate by TP: TP Ref #: OMA-TP-2009-0217-INP_DM_V1_3_RD_and_AD_for_Candidate_Approval

Appendix B. Management Authority Diagram and Text (Informative)

B.1 Architectural Diagram

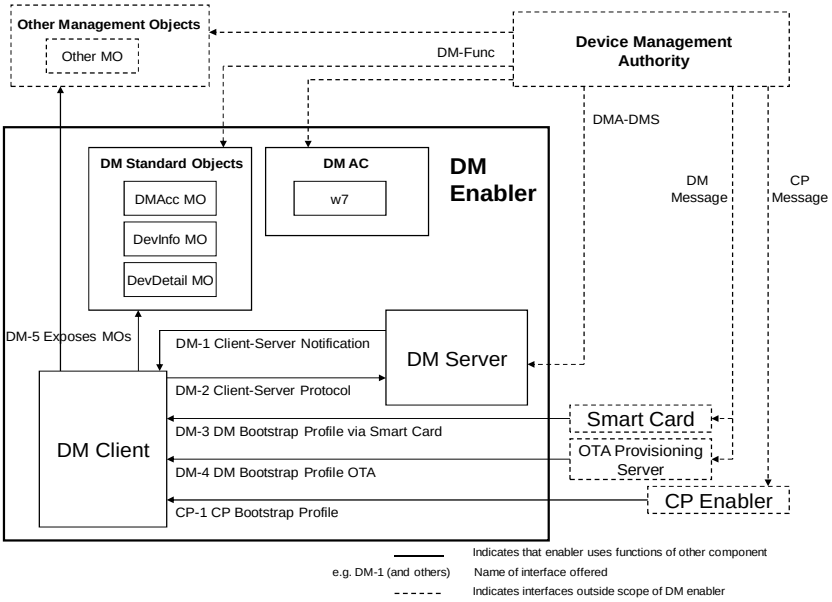

Figure 2: Device Management Architecture using interfaces

B.2 Additional Interfaces

B.2.1 DM-5 DM Exposes Management Objects

The MO schema are exposed by the {DM Client} through its device management tree. The parameters exposed through the MO schema may be accessed by the {DM Server} through the Target LocURI element of the DM representation protocol in DM messages.

B.2.2 DM-Func DM Functions

The {Standard Management Objects} represent interfaces to the Device's {DM Client} configuration and the Device's DM-related information which may be targeted by a {Device Management Authority} to perform Device Management Functions. The functions available depend upon the DM Standard Object specifications (e.g. Get, Replace, Add, Delete, Atomic, and Sequence), the access rights assigned to specific parameters for a given Device Management Authority, and on the specific device implementation.

B.2.3 DMA-DMS Interface

The interfaces between a Device Management Authority's line-of-business systems and a Device Management Server are out of scope. For purposes of illustration, this interface allows the Device Management Authority to submit device management requests to the DM Server and to be apprised of results and device-generated alerts received by the server from the DM Client. For purposes of this reference architecture description, readers should assume that an implementation-specific interface to the DM Server is used by the Device Management Authority to submit DM commands and analyze results returned by the DM Client.

B.2.4 DM Message

The {Device Management Authority} sets the initial provisioning information into the DM Message that can then be used by the {DM Client}. The details of the DM Message are decided by the {Device Management Authority} and typically relate to information necessary for the {DM Client} to connect to the {DM Server}.

B.2.5 CP Message

The {Device Management Authority} sets the initial provisioning information into a file that can then be used by the {CP Enabler}. The details of the CP Message are decided by the {Device Management Authority} and typically relate to information necessary for the {DM Client} to connect to the {DM Server}.

B.3 Data Objects

B.3.1 Management Objects

B.3.1.1 DMAcc Management Object

Standardized interface to the DM Account configuration – the information required for the {DM Client} to communicate with the {DM Server}. Exposed through the {DM Client} for authorized access by {Device Management Authority} utilizing {DM Server} communicating over {DM-2}.

B.3.1.2 DevInfo Management Object

Standardized interface to the basic Device information. Exposed through the {DM Client} for authorized access by {Device Management Authority} utilizing {DM Server} communicating over {DM-2}. This information is also transmitted by the {DM Client} to the {DM Server} during session establishment.

B.3.1.3 DevDetail Management Object

Standardized interface to the detailed Device information. Exposed through the {DM Client} for authorized access by {Device Management Authority} utilizing {DM Server} communicating over {DM-2}.

B.3.2 Application Characteristics

B.3.2.1 w7 Application Characteristic

Standardized interface to the DM Account configuration. Exposed by the {DM Client} to the {Device Management Authority} utilizing {CP Enabler} communicating over {CP Message}.

B.4 Flows

This flow describes the normal device management flow. The DM Client and the DM Server will exchange authentication, and then the DM Server will send commands to the DM Client.

Figure 2 shows the normal flow for this scenario:

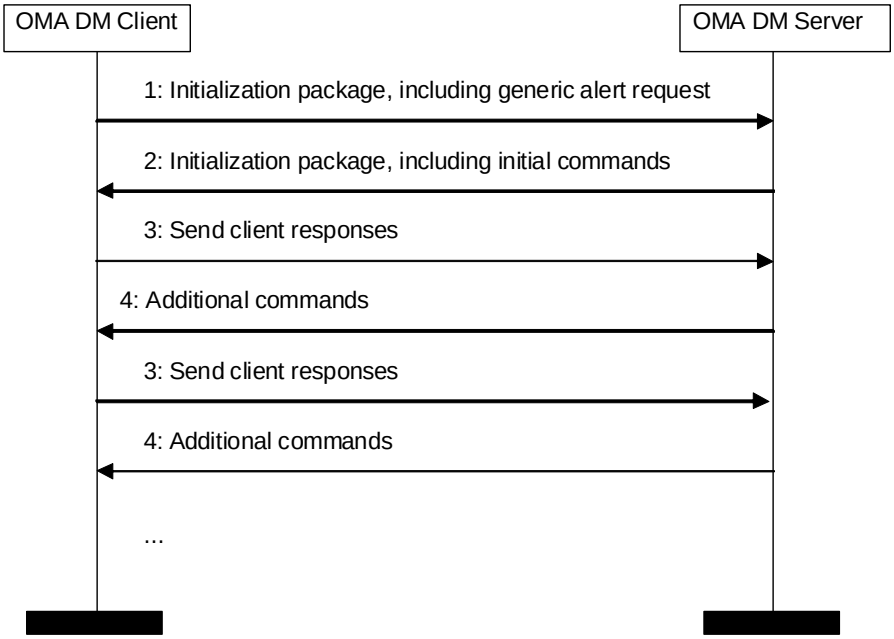

Figure 3: Normal Device Management Flow

The detailed flow is as the following:

Step 1: The DM Client sends the initialization package to the DM Server, including authentication information, device information and any generic alert requests.

Step 2: The DM Server authenticates the DM Client, analyses the generic alert requests (if any) and determines an appropriate set of commands. Then the DM Server sends the initialization package to the DM Client, including authentication information, and an initial set of commands.

Step 3: The DM Client performs the commands, and sends back responses to the commands.

Step 4: The DM Server reviews the command responses and sends the DM Client additional commands.

Step 3 and 4 repeat until there are no more commands in Step 4.

B.4.1 Alternative flow 1

Optionally, the DM Server may send an out-of-band notification to the DM Client. Upon receiving the notification, the DM Client should connect to the server as soon as practical.

B.4.2 Alternative flow 2

Optionally, the DM Server may send a bootstrap message to the DM Client. The DM Client, upon receiving the bootstrap message, should contact the server as soon as practical.

B.4.3 Alternative flow 3

Optionally, the DM Client may send a Generic Alert as part of Step 3.

B.5 Security Considerations

DM 1.3 requires a high level of security, due to the data that is being handled. If a DM Client were to be configured by a rogue DM Server, it is possible for the device to be ruined. If a rogue DM Client were to be configured by a DM Server, it is possible for the data from that DM Client to propogate into the network (if the DM Client were masquerading as another device).

In the end, the service provider:

- provides mutual authentication of the DM Client and DM Server.
- does not allow un-authorized DM Servers or DM Clients to communicate.
- provides secure transfer of exchanged data to and from the DM Client.
- provides for data integrity between DM Client and DM Server.
- provides for confidentiality of personal data or data related to the owner of the device.