Device Management Architecture

Approved Version 1.3 – 24 May 2016

Open Mobile Alliance

OMA-AD-DM-V1_3-20160524-A

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

[DM12] "OMA Device Management Enabler Release Definition", Version 1.2, Open Mobile Alliance™, OMA-ERELD-DM_V1_2.
URL:http://www.openmobilealliance.org/

[DMDICT] "OMA Device Management Dictionary, Version 1.0". Open Mobile Alliance™. OMA-SUP-DM_Dictionary-v1_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

[SIPPUSH] "Push using SIP". Open Mobile Alliance™. OMA-TS-SIP_Push-V1_0.
URL:http://www.openmobilealliance.org

2.2 Informative References

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

[DM-RD] "Device Management Requirements", Open Mobile Alliance™, OMA-RD-DM-V1_3,
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

Kindly consult [DMDICT] for all definitions used in this document.

3.3 Abbreviations

Kindly consult [DMDICT] for all abbreviations used in this document.

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 mechanism 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 reuses the architecture from OMA DM 1.2. It does introduce new notification, transport protocols and a new DM Server to DM Server interface for delegation.

5. Architectural Model

5.1 Dependencies

DM 1.3 will have the same dependencies as the DM 1.2 enabler [DM12].Additionally, DM 1.3 optionally depends on OMA SIP Push enabler [SIPPUSH].

5.2 Architectural Diagram

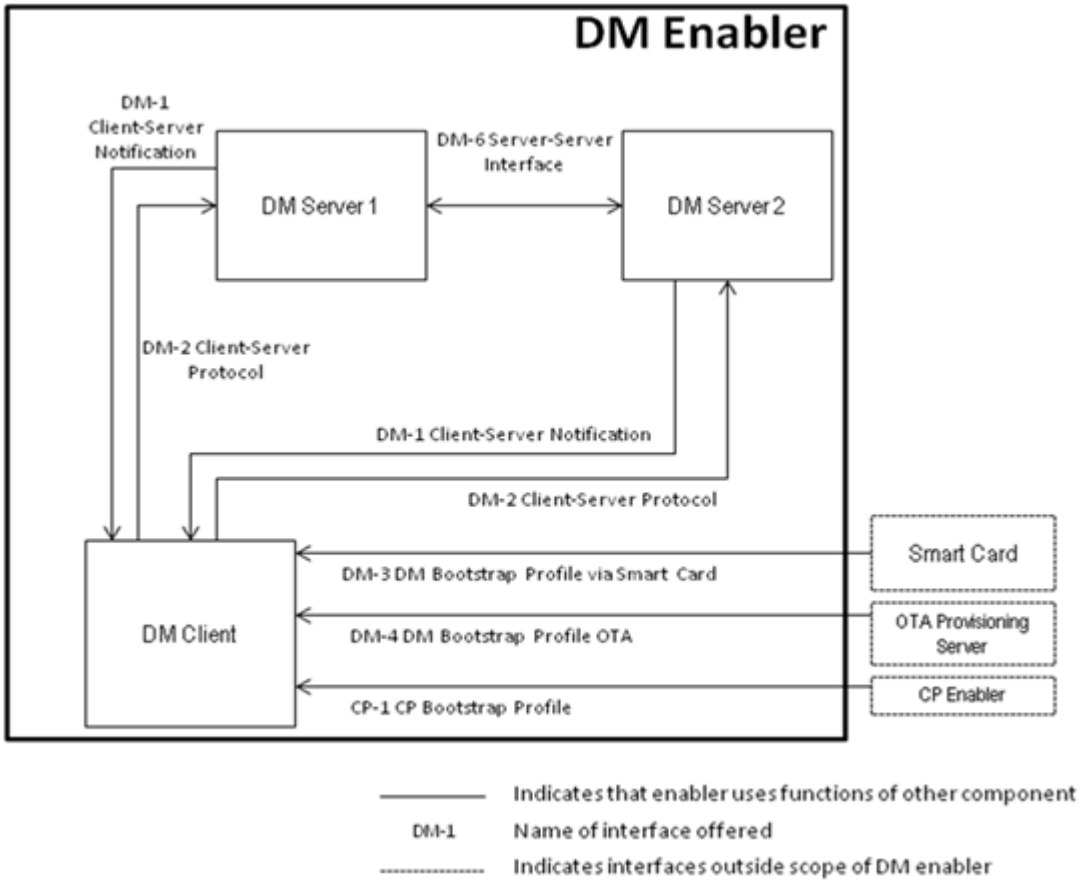

Figure 1: Device Management Architecture using interfaces

5.3 Functional Components and Interfaces

In this chapter, the description of components and interfaces represented in Figure 1 is provided. For more detailed information please refer to Appendix B.

5.3.1 Components

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 DM Client-Server Notification

This provides an interface over which the DM Servers may send device management notification to the DM 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 DM Client-Server Protocol

This provides an interface over which the DM Servers may send DM Commands to the DM Clients and the DM Clients may return status and alerts to the DM 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.

5.3.2.6 DM-6 DM Server-Server Interface

This provides an interface over which DM Servers may send management commands to other DM Servers and receive responses from other DM Servers for delegation. This is an interface that is bearer neutral and offers many standardized bindings including HTTP and HTTPS.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-AD-DM-V1_3-20160524-A	24 May 2016	Status changed to Approved by TP TP Ref # OMA-TP-2016-0041R01-INP_DM_V1_3_ERP_for_final_Approval

Appendix B. Flows (Informative)

B.1 Normal DM Flow

This flow describes the normal DM 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:

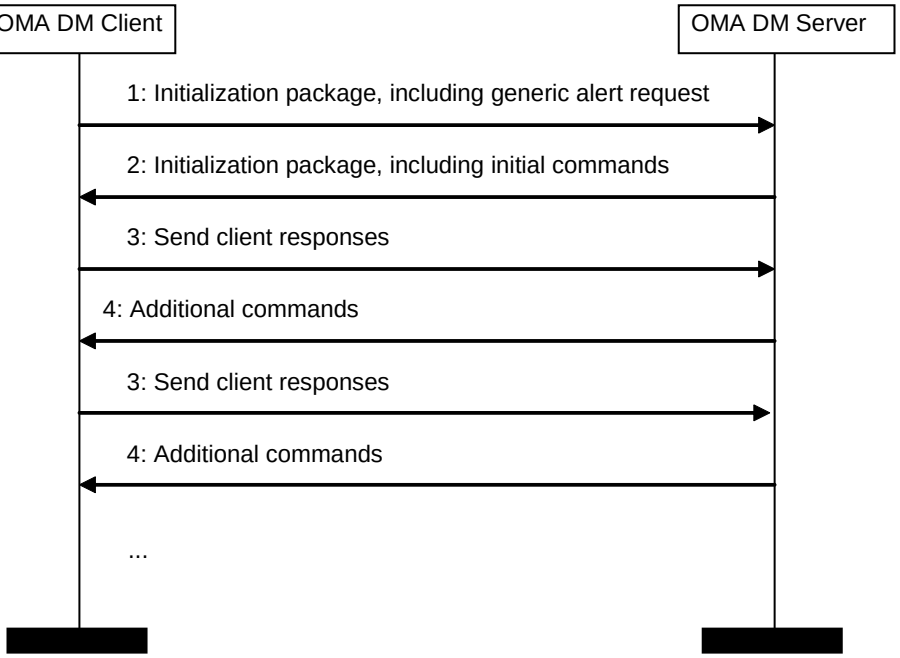

Figure 2: Normal Device Management Flow

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 are to be repeated until there are no more commands in the Step 4.

B.1.1 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 DM Server as soon as practical.

B.1.2 Alternative flow 2

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

B.1.3 Alternative flow 3

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

Appendix C. Management Authorities in DM (Informative)

C.1 Architectural Diagram

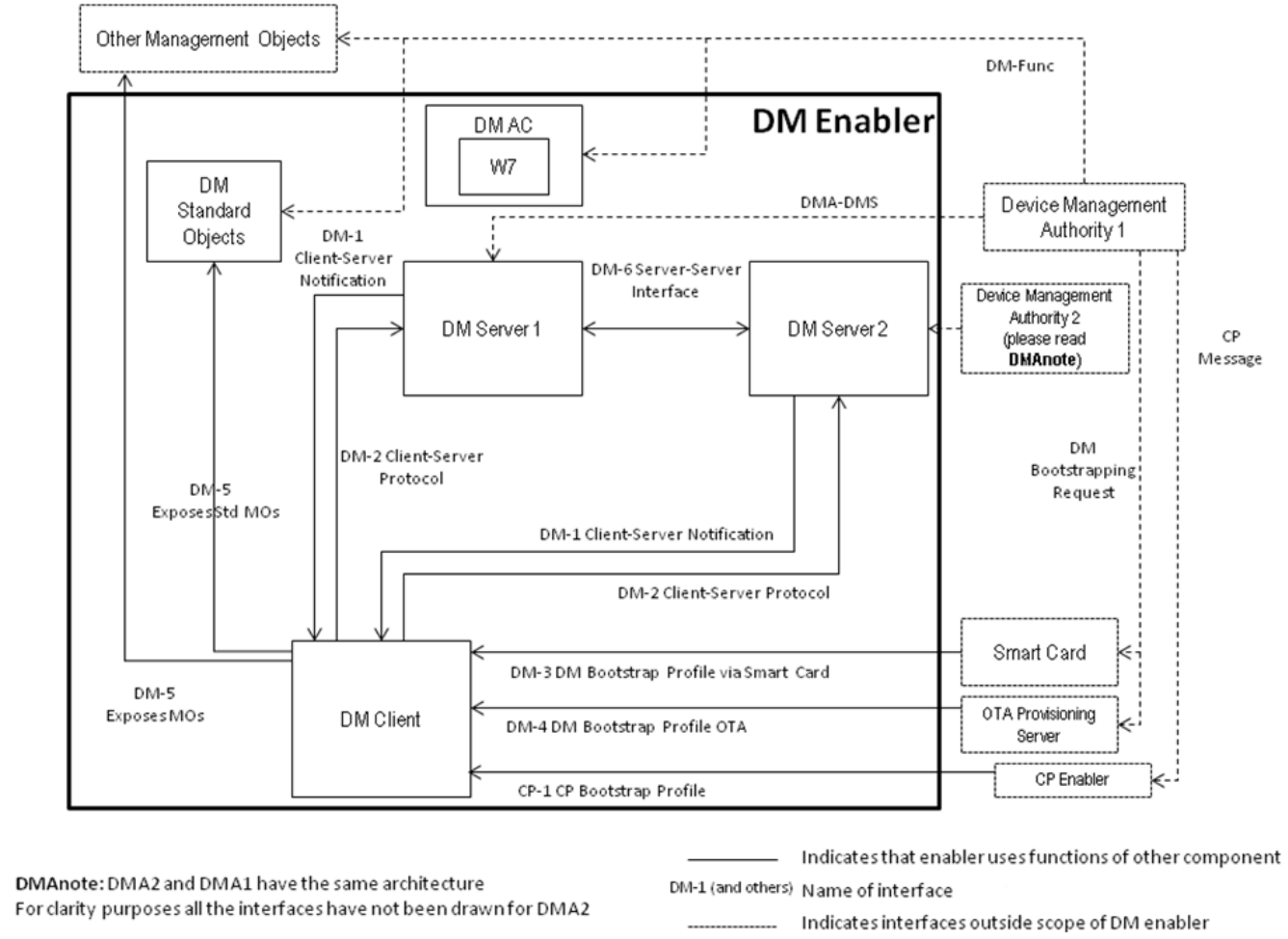

Figure 3: Device Management Architecture using interfaces

C.2 Additional Interfaces

This chapter defined interfaces in addition to those described in chapter 5.3.2

C.2.1 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 DM Authority to perform DM 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.

C.2.2 DMA-DMS Interface

The interfaces between a DM Authority's line-of-business systems and a DM Server are out of scope. For purposes of illustration, this interface allows the DM Authority to submit DM 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 DM Authority to submit DM commands and analyze results returned by the DM Client.

C.2.3 DM Bootstrapping Request

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

C.2.4 CP Message

The DM 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 DM Authority and typically relate to information necessary for the DM Client to connect to the DM Server.

C.2.5 DM-5 DM Management Objects Exposure

The MO schemas are exposed by the DM Client through its device management tree. Standard Management Objects are defined by the DM enabler. Those MOs are DMAcc, DevInfo, DevDetail and Inbox as defined below.

C.3 Data Objects

C.3.1 Standard Management Objects

C.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 DM Authority utilizing DM Server communicating over DM-1.

C.3.1.2 DevInfo Management Object

Standardized interface to the basic device information exposed through the DM Client for authorized access by DM Authority utilizing DM Server communicating over DM-1. This information is also transmitted by the DM Client to the DM Server during session establishment.

C.3.1.3 DevDetail Management Object

Standardized interface to the detailed device information exposed through the DM Client for authorized access by DM Authority utilizing DM Server communicating over DM-1.

C.3.2 Application Characteristics

C.3.2.1 w7 Application Characteristic

Standardized interface to the DM Account configuration, exposed by the DM Client to the DM Authority utilizing CP Enabler communicating over CP Message.

C.4 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.

DM 1.3 supports DM Server to DM Client and DM Server to DM Server mutual authentications for delegation.