Gateway Management Object Technical Specification

Approved Version 1.1 – 25 Jul 2017

Open Mobile Alliance

OMA-TS-GwMO-V1_1-20170725-A

Contents

Figures

1. Scope

This technical specification describes Management Objects and Generic Alerts that are needed to provide the DM Gateway functionality, as defined in [DMDICT].

2. References

2.1 Normative References

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

[DMNOTI] "OMA Device Management Notification Initiated Session", Version 1.3, Open Mobile Alliance™, URL:http://www.openmobilealliance.org

[DMPRO] "OMA Device Management Protocol", Version 1.3. Open Mobile Alliance™. OMA-TS-DM_Protocol-V1_3, URL:http://www.openmobilealliance.org

[DMREPPRO] "OMA Device Management Representation Protocol, Version 1.3", Open Mobile Alliance™. OMA-TS-DM_RepPro-V1_3, URL:http://www.openmobilealliance.org

[DMSTDOBJ] "OMA Device Management Standardized Objects". Version 1.3, Open Mobile Alliance™, URL:http://www.openmobilealliance.org

[DMTND] "OMA Device Management Tree and Description", Version 1.3, Open Mobile Alliance™, OMA-TS-DM_TND-V1_3, URL:http://www.openmobilealliance.org/

[GwMO_AD_v1.1] "GwMO Architecture", Version 1.1, Open Mobile Alliance™, OMA-AD-GwMO-V1_0, URL:http://www.openmobilealliance.org/

[GwMO_RD_v1.1] "GwMO Requirements", Version 1.1, Open Mobile Alliance™, OMA-RD-GwMO-V1_0, URL:http://www.openmobilealliance.org/

[GwMO_ZigBeeMO_TS_v1_0] "Management Objects for ZigBee Devices, Version 1.0 ", Open Mobile Alliance™, OMA-TS-DM-GwMO_ZigBeeMO -V1_0, URL:http://www.openmobilealliance.org/

[OMGIDL] "CORBA 3.1 – OMG IDL Syntax and Semantics chapter", Object Management Group, January 2008, URL:http://www.omg.org/cgi-bin/doc?formal/08-01-04.pdf

[PushMsg] "Push Message Specification", Version 2.3, Open Mobile Alliance™ OMA-TS-Push_Message-V2_3, URL: http://www.openmobilealliance.org/

[PushOTA] "Push Over the Air", Version 2.3, Open Mobile Alliance™, OMA-TS-PushOTA-V2_3, 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

[RFC5234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. January 2008, URL:http://www.ietf.org/rfc/rfc5234.txt

[SCRRULES] "SCR Rules and Procedures", Version 1.0, Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:http://www.openmobilealliance.org/

[SIPPush] "Push using SIP", Open Mobile Alliance™, Version 1.0, OMA-TS-SIP_Push-V1_0, URL:http://www.openmobilealliance.org/

[ZigBee2007] "ZigBee Specification 2007", ZigBee Alliance, Version 053474r17, URL: http://www.zigbee.org/Specifications/ZigBee/download.aspx

2.2 Informative References

[DLOTA] "Generic Content Download Over The Air Specification Version 1.0", Open Mobile Alliance™, OMA-Download-OTA-v1_0, URL:http://www.openmobilealliance.org/

[DMARCH] "OMA Device Management Architecture", Version 1.3, Open Mobile Alliance™ OMA-AD-DM-V1_3, URL:http://www.openmobilealliance.org/

[DMBOOT] "OMA Device Management Bootstrap", Version 1.3, Open Mobile Alliance™, OMA-TS-DM_Bootstrap-V1_3, URL:http://www.openmobilealliance.org/

[SCOMO-TS] "Software Component Management Object", Version 1.0, Open Mobile Alliance™, OMA-TS-DM-SCOMO-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

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

The OMA DM [DMPRO] protocol is used for the remote management of devices. In most instances, the OMA DM Server and the OMA DM Client communicate with each other directly. However, direct communication between the DM Server and the DM Client is not always possible, nor desirable, due to inaccessibility of devices behind a firewall or devices supporting a management protocol other than OMA DM. This specification provides a framework for OMA DM to manage devices indirectly (that is, through a DM Gateway [DMDICT], as illustrated in Figure 1).

The DM Gateway supports various modes of operation. The precise role played by the DM Gateway in a management session involving a DM Server and an End Device depends upon the mode of operation of the DM Gateway. In some instances the DM Gateway is managed by an OMA DM Server, and in turn, the DM Gateway manages other End Devices under it. In other instances, the DM Gateway merely enables a DM Server to communicate with an otherwise unreachable End Device. The DM Gateway's modes of operation are defined in Section 5.1.

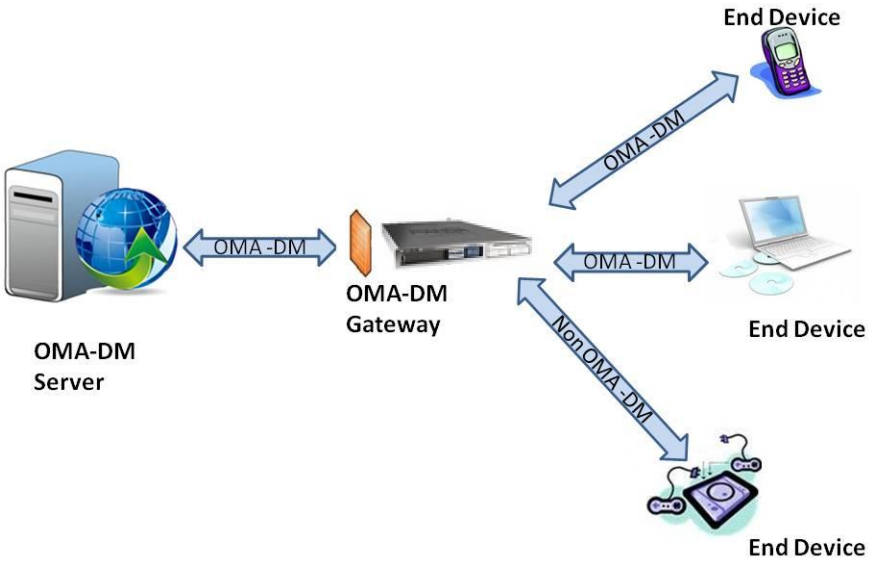

Figure 1: Device Management Via a DM Gateway

4.1 Version 1.0

Version 1.0 introduces the OMA Gateway MO enabler. This version defines the following MOs:

- Device Inventory MO
- Gateway Config MO
- Fanout MO
- Image Inventory MO
- End Device Trigger MO

This version also defines the following alert types:

- Device Attach Alert
- Device Detach Alert
- Fanout Result Aggregation Alert
- Fanout Completion Status Alert
- Bootstrapped DMS List Alert
- Associated Gateway Alert
- Image Ready Alert

Additionally, the specification provides design guidelines for the adaptation of non OMA DM protocols to OMA DM.

4.2 Version 1.1

Version 1.1 introduces the following new functionalities and enhancements for OMA Gateway MO 1.0:

- Support for enhanced Adaptation Mode

  GwMO V1.0 has presented Adaptation Mode of operation, but the mapping rules between OMA DM protocol and the non-OMA DM management protocol of the End Devices were outside the scope of V1.0. Version 1.1 presents the mapping rules and adaptation approaches for managing devices implementing protocols other than OMA DM such as ZigBee. Refer to Section 8.4 for description and framework for realizing the Adaptation Mode and GwMO architecture section 5.2.3 in [GwMO_AD_v1.1] for supporting adaptation functionality. Please refer to [GwMO_ZigBeeMO_TS_v1_0] for information about adaptation mode for ZigBee devices and the definition of the ZigBee Management Objects.

- Support for Hierarchical Architecture

  Version 1.1 introduces Hierarchical Architecture as a multi-level tree structure composed of DM Server, DM Gateways and End Devices. Refer to Sections 6.1 and 8.1.2 for ComponentType node and description Inventory Update Flow for Hierarchical Architecture. Hierarchical Architecture may address larger networks, as in M2M, with a greater number of End Devices in a very scalable and flexible way:

  - Scalability: in V1.0 with a one-level Gateway structure the maximum number of supported devices is limited by the number of connections held by the Gateway. In multi-level Hierarchical Architecture maximum number of supported devices increases exponentially.
  - Flexibility: more convenient management topologies may be designed based on geographic, functional or organizational criteria. Example: Gateways dedicated to specific locations (e.g. building, building floor, block, office), specific functions (e.g. energy consumption metering, security, cooling-heating systems) or organization (department, division, branches). For example, company may have a dedicated Gateway for energy devices and another one for security devices and a top level Gateway aggregating those Gateways and connected to company DM Server

- Security Enhancements for Proxy Mode Operations

  Resolves V1.0 security vulnerability with GwMO V1.0 operating in Proxy mode: the credentials and ACL from the session between DM Server and DM Gateway were not propagated to the session between the DM Gateway and the End Device. Refer to Sections 6.2 and 8.3.4 for further information on secure proxy mode operation and Privileges sub-tree in Gateway Config MO.

5. Overview of DM Gateway Functionality (Normative)

This section provides an overview of the DM Gateway functionality. The DM Server and DM Gateway interconnect over the underlying DM-1 interface, as defined in [DMARCH], and use the GwMO-1 and GwMO-2 interfaces defined in [GwMO_AD_v1.1].

The DM Gateway maintains Management Objects to support:

- The collection of device identifier and device address information relating to End Devices attached to the DM Gateway and making this information available to a DM Server (that is, Device Inventory MO)
- The organization of End Devices into Device groups, the configuration of the Bootstrap Server URL and controlling the reporting of alerts pertaining to the End Devices to the DM Server (that is, Gateway Config MO)
- The distribution of DM commands to multiple End Devices that are attached to the DM Gateway (that is, Fanout MO)
- The distribution of DM Notification to End Device(s) that are attached to the DM Gateway (that is, End Device Trigger MO)
- The local storage of the software images for End Devices, for efficient software distribution (that is, Image Inventory MO)

5.1 Gateway Modes of Operation

The DM Gateway provides three modes of operation: Transparent, Proxy, and Adaptation. These modes of operation allow interconnection and management operations between a DM Server and End Devices, as described in the following sub-sections. The DM Gateway MUST support Transparent and Proxy mode.

The DM Gateway can operate in different modes for different devices simultaneously.

5.1.1 Transparent Mode

In this mode, the DM Gateway assists the DM Server in sending a DM Notification [DMNOTI] to the End Device(s) behind the DM Gateway. The DM Gateway forwards the DM Notification to the End Device(s). The DM Gateway does not participate in the management session that gets established between the DM Server and the End Device(s) after the delivery of the DM Notification to the End Device(s).

5.1.2 Proxy Mode

In this mode, the DM Gateway manages End Device(s) behind the DM Gateway on behalf of the DM Server over the OMA DM protocol. Two related DM sessions are established: one is between the DM Server and the DM Gateway; the other is between the DM Gateway and the End Device(s).

The DM Server sends management commands to the DM Gateway and the DM Gateway forwards the management commands to the targeted End Device(s), within a new message generated by the DM Gateway. After the DM Gateway receives the responses sent by the End Device(s), the DM Gateway may notify the DM Server about the command completion status, or it may send the aggregated response of the command, from multiple End Devices, to the DM Server. The command results are available for retrieval through the Fanout MO interface on the DM Gateway.

5.1.3 Adaptation Mode

In this mode, the DM Gateway manages End Device(s) behind the DM Gateway, on behalf of the OMA DM Server over a non-OMA DM protocol. When operating in the Adaptation Mode, the DM Gateway is expected to make a best effort to translate between the OMA DM commands and the other management protocol.

6. Management Objects (Normative)

This section describes the various Management Objects (MOs) that have been defined for realizing the DM Gateway [DMDICT] functionality. The MOs conform to the Management Object definition and description convention, as outlined in the [DMTND] specification.

6.1 Device Inventory MO

6.1.1 MO Description

The Device Inventory MO resides in the Management Tree [DMTND] of the DM Gateway and it maintains a list of devices in the network that are managed through the DM Gateway.

This MO is updated when the DM Gateway becomes aware of a new End Device in the network or the DM Gateway becomes aware that a previously subtending End Device is no longer present in the network.

Figure 2 gives the pictorial description of the Device Inventory MO. The description of the various nodes within this MO is given below.

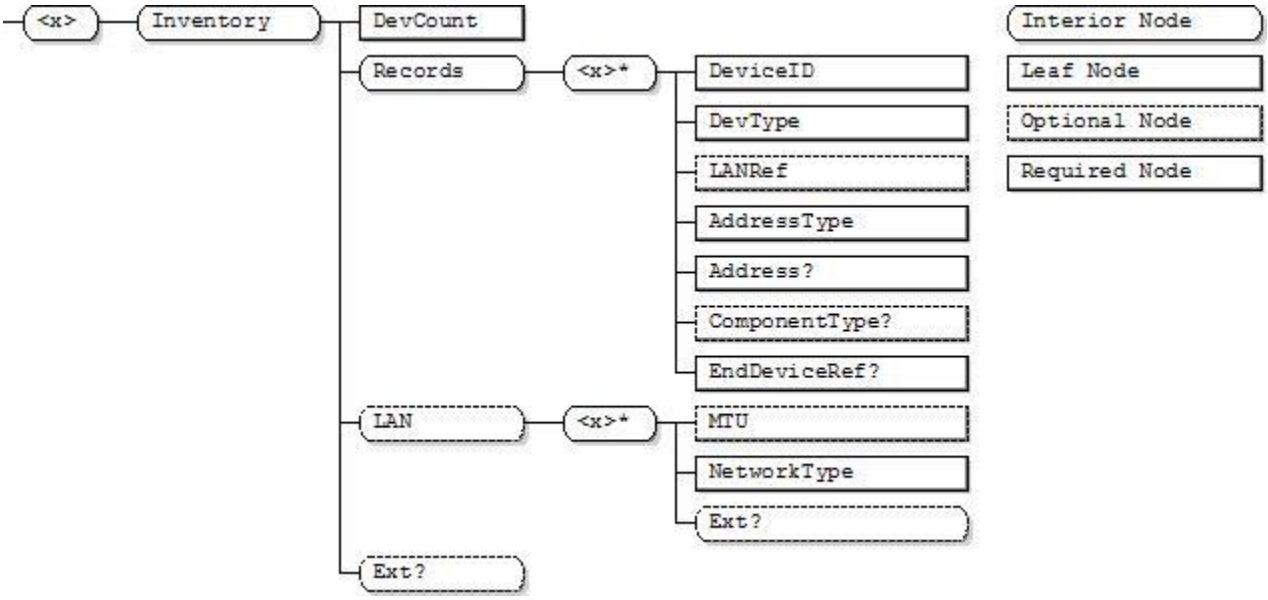

Figure 2: Device Inventory MO

<x>

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This placeholder node is the root node for the Device Inventory MO. The parent node of this node defines the location of this MO in the DM Gateway's Management Tree.

The Management Object Identifier for the Device Inventory MO MUST be: “urn:oma:mo:oma-gwmo-deviceinventory:1.1”.

<x>/Inventory

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is the parent node for all device inventory information.

<x>/Inventory/DevCount

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get, No Replace

This leaf node gives the number of End Devices that are managed through the DM Gateway. This value MUST only be set by the DM Gateway.

<x>/Inventory/Records

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get, No Replace

This interior node is the parent node for all device inventory entries.

<x>/Inventory/Records/<x>

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get, No Replace

This placeholder node contains the device inventory entry for a specific End Device that is subtending from the DM Gateway.

<x>/Inventory/Records/<x>/DeviceID

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get, No Replace

The value of this leaf node provides the public identity of the Device. This identifier is used by the DM Server to indicate to the DM Gateway the target End Device to be managed. For End Devices supporting OMA DM, this value SHOULD be the same as the value of the *DevId* node in the DevInfo MO [DMSTDOBJ]. If the End Device does not support the notion of device identity, this value MUST be assigned by the DM Gateway and it MUST be unique within the Management Tree of the DM Gateway.

<x>/Inventory/Records/<x>/DevType

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get, No Replace

The value of this leaf node indicates the Device type. For End Devices supporting OMA DM, this value MUST be the same as the value of the *DevType* node in the DevDetail MO [DMSTDOBJ].

<x>/Inventory/Records/<x>/LANRef

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	chr	Get, No Replace

The value of this leaf node indicates the nodename that provides information about the LAN in which the End Device is deployed. The value of this node MUST be matched with one of the child nodes of the '*<x>/Inventory/LAN/<x>*' node.

If the Gateway supports this node, it MUST also support <x>/Inventory/LAN node.

<x>/Inventory/Records/<x>/AddressType

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get, No Replace

The value of this leaf node specifies the address type of the End Device. Valid values for this node MUST be one of the following:

Value	Meaning	Description
0	Non-Routable	The address of the End Device is non-routable. The DM Server cannot access the End Device directly.
1	IPV4	The address of the End Device is a publicly routable IPv4 address.
2	IPV6	The address of the End Device is a publicly routable IPv6 address.
3	GRUU	The address of the End Device is a GRUU address. The DM Notification will be sent as a connectionless SIP MESSAGE or a connection oriented MSRP message [SIPPush].

<x>/Inventory/Records/<x>/Address

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get, No Replace

The value of this leaf node provides the public routable address of the End Device. This node MUST NOT be present if the value of the sibling *AddressType* node is 0 (that is, Non-Routable).

<x>/Inventory/Records/<x>/Mode

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get, No Replace

The value of this leaf node indicates the operation mode of the DM Gateway for the End Device. Valid values for this node MUST be one of the following:

Value	Meaning
1	The Gateway is operating only in the Transparent Mode for the End Device.
2	The Gateway is operating only in the Proxy Mode for the End Device.
3	The Gateway is operating in both the Transparent Mode and in the Proxy Mode for the End Device.
4	The Gateway is operating only in the Protocol Adaptation Mode for the End Device.

<x>/Inventory/Records/<x>/ComponentType

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

If Hierarchical Architecture is supported, the value of this leaf node indicates the type of the GwMO component directly associated to this DM Gateway:

Value	Meaning	Description
0	End Device	An End Device is directly associated to this DM Gateway.
1	DM Gateway	A child DM Gateway is directly associated to this DM Gateway.
>= 2	Reserved for future use	Reserved.

The value of this node MAY be used to support DM Gateway features configuration and Device Group in Section 6.2 Gateway Config MO. If this node is absent, it is assumed the ComponentType is an End Device (i.e. value 0).

<x>/Inventory/Records/<x>/EndDeviceRef

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node specifies a reference to the node which represents a non-OMA End Device in any other MO.

This node is required when the DM Gateway operates in Adaptation Mode and plays the role of the Origin Server by hosting the Management Tree of the End Device. (See section 8.4.3 and Figure 22: GwMO Adaptation Mode Realization with Gateway as Origin Server

.) In this case, the DM Gateway hosts in its Management Tree a representation of a non-OMA DM End Device apart from that of the DM Gateway itself. A representation of a non-OMA DM End Device SHOULD be linked to an entry in the Device Inventory MO. This node works as a link between them.

For ZigBee End Devices, refer to [GwMO_ZigBeeMO_TS_v1_0].

It is expected that a URI to the target node is specified, but other implementation-specific form of a link to the target node MAY be referenced.

<x>/Inventory/LAN

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get, No Replace

This interior node is for storing information regarding the local area network in which the End Device is deployed.

<x>/Inventory/LAN/<x>

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get, No Replace

This placeholder node is for storing information regarding the local area network in which the End Device is deployed. The name of this node is referred by the value of the '*<x>/Inventory/Records/<x>/LANRef* ' node.

<x>/Inventory/LAN/<x>/MTU

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	int	Get, No Replace

The value of this node indicates the Maximum Transmission Unit (MTU) size, in bytes, for the network between the DM Gateway and the End Device.

<x>/Inventory/LAN/<x>/NetworkType

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get, No Replace

The value of this node indicates the network type which is used between Gateway and End Device.

Value	Meaning	Description
0	Unknown	Network Type is unknown
1	Wired IP	Wired IP based network
2	Bluetooth	Bluetooth communication link
3	ZigBee	ZigBee communication link
4	WiFi	Wireless IP based network
5	6LoWPAN	IPv6 over Low power Wireless Personal Are Network

<x>/Inventory/LAN/<x>/Ext

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node is for vendor-specific extensions to store the LAN related information

<x>/Inventory/Ext

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node is for vendor-specific extensions to the Device Inventory MO.

6.2 Gateway Config MO

6.2.1 MO Description

The Gateway Config MO resides in the Management Tree [DMTND] of the DM Gateway and it maintains information regarding the handling of different types of End Devices by the DM Gateway and provides secure management operations on end device or group. This MO contains the following sub-trees:

- DevTypeInfo: This sub-tree is used to hold control information for reporting the Device Attach/Detach alerts and providing bootstrapping information depending on the device type of the End Devices.
- Config: This sub-tree is used by the DM Server to configure the DM Gateway for features such as the reporting of alerts that the DM Gateway might send to the DM Server. It is also used to store End Device credentials on the DM Gateway for those End Devices that require DM Server assisted bootstrapping [section 8.5.1.1].
- DevGroup: This sub-tree is used to assign attached End Devices into groups, which can then be addressed for command fanouts or notification fanouts.

Priviliges: This sub-tree is used to provide the secure management operations in proxy mode. This contains access rights on a certain resources in device. This provides privilege permissions for the servers to send management commands on the group of devices or single device. For any management operation from server on the group of end devices or end device, DM Gateway would check this permissions for authorisation of Originating server's Management Command. This also used to

add or change the rights for other servers by receiving a device management command to Privileges from primary server to add/modify an ACL for secondary server who doesn't have access right on a certain resource in a device or device group on the DM Gateway.

Figure 3 gives the pictorial description of the Gateway Config MO. The description of the various nodes within this MO is given below.

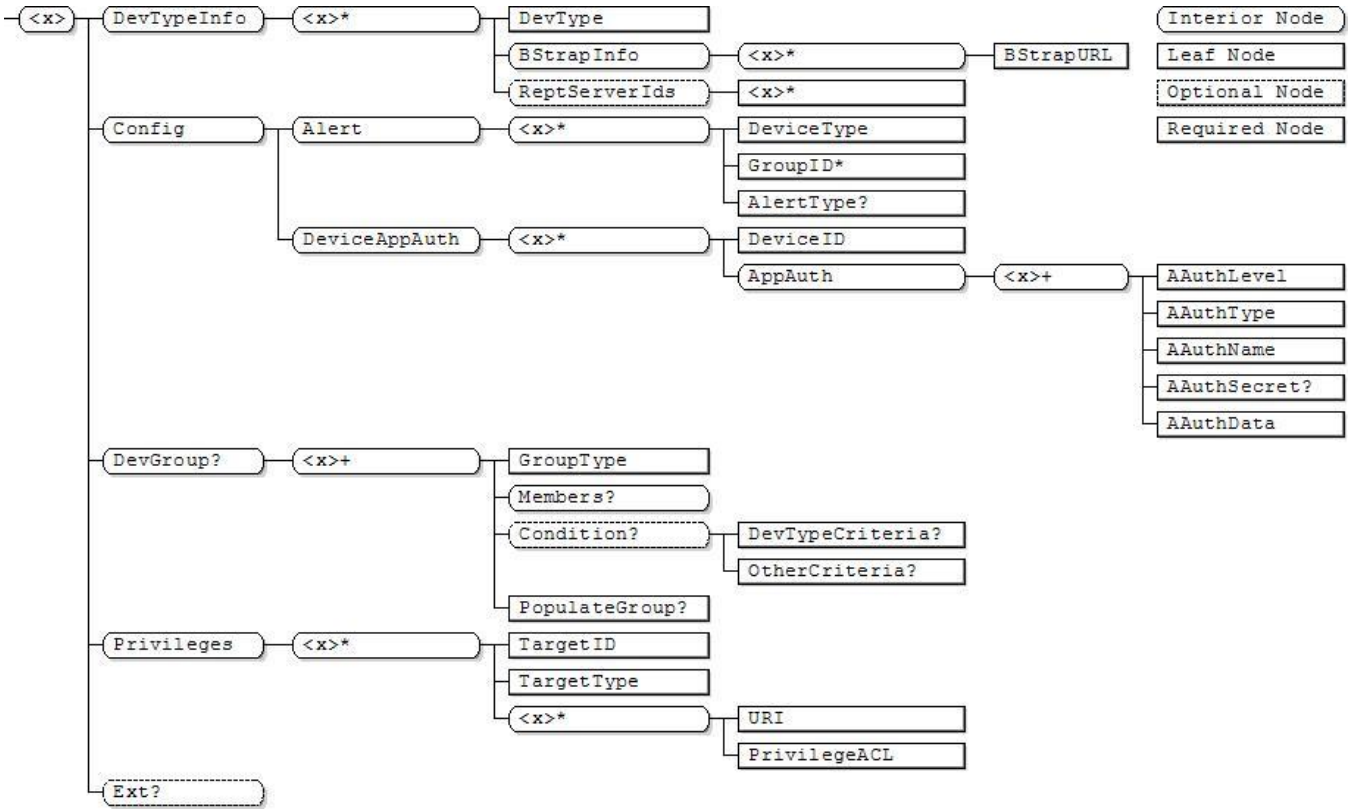

Figure 3: Gateway Config MO

<x>

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This placeholder node is the root node for the Gateway Config MO. The parent node of this node defines the location of this MO in the DM Gateway's Management Tree.

The Management Object Identifier for the Gateway ConfigMO MUST be: "urn:oma:mo:oma-gwmo-config:1.1".

<x>/DevTypeInfo

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node contains all of the information pertaining to the handling of different types of End Devices by the DM Gateway.

<x>/DevTypeInfo/<x>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get, Add, Delete

This placeholder node contains information about the handling of one particular type of End Device by the DM Gateway.

<x>/ DevTypeInfo/<x>/DevType

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get, Replace

The value of this leaf node specifies the device type.

<x>/DevTypeInfo/<x>/BStrapInfo

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node contains all bootstrapping records that are needed for client-initiated bootstrap for the device type.

<x>/DevTypeInfo/<x>/BStrapInfo/<x>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get, Add, Delete

This placeholder node contains one bootstrapping record that has all of the information for client-initiated bootstrap for the device type.

<x>/DevTypeInfo/<x>/BStrapInfo/<x>/BStrapURL

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get, Replace

The value of this leaf node indicates the URL of the Bootstrap Server for the device type.

<x>/DevTypeInfo/<x>/ReptServerIds

Status	Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node contains information about all DM Servers to which the DM Gateway MUST report alerts pertaining to End Devices of the specified device type.

<x>/DevTypeInfo/<x>/ReptServerIds/<x>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	null	Get, Add, Delete, Replace

The name of this leaf node is the identifier of one DM Server to which the DM Gateway MUST report alerts pertaining to End Devices of the specified device type. The DM Gateway MUST have been previously bootstrapped to this DM Server.

<x>/Config

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node contains the configuration parameters for the DM Gateway.

<x>/Config/Alert

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is the root node for all of the configuration parameters dealing with the reporting of Generic Alerts pertaining to End Devices. If this node has no children, the DM Gateway reports all Generic Alerts pertaining to End Devices to the DM Server(s).

<x>/Config/Alert/<x>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This placeholder node groups the configuration parameters for the reporting of Generic Alerts based on some criteria (for example, alert type, device type, or device group). This node MUST contain either the *DevType* child node or the *GroupID* child node, but not both.

<x>/Config/Alert/<x>/DeviceType

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

The value of this leaf node specifies the device type. Note that for OMA DM devices, the device type is determined by the *DevType* node in the DevDetail MO [DMSTDOBJ].

This node is mutually exclusive with its sibling *GroupID* node.

<x>/Config/Alert/<x>/GroupID

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	chr	Get

The value of this leaf node specifies the GroupID, which is specified in the DevGroup sub-tree of this MO.

This node is mutually exclusive with its sibling *DevType* node.

<x>/Config/Alert/<x>/AlertType

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

The value of this leaf node specifies the Generic Alert type. Absence of this node implies all Generic Alert types.

<x>/Config/DeviceAppAuth

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This node specifies authentication information for hosting End Devices that require DM Server assisted bootstrapping [section 8.5.1.1].

<x>/Config/DeviceAppAuth/<x>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This placeholder node stores the authentication information for a particular End Device.

<x>/Config/DeviceAppAuth/<x>/DeviceID

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This node specifies the Device ID associated with the AppAuth setting.

<x>/Config/DeviceAppAuth/<x>/AppAuth

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This node specifies authentication information for the End Device whose identifier is the value of the sibling *DeviceID* node.

<x>/Config/DeviceAppAuth/<x>/AppAuth/<x>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This interior node acts as a placeholder for one or more authentication settings.

<x>/Config/DeviceAppAuth/<x>/AppAuth/<x>/AAuthLevel

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This node specifies the authentication level. See details in [DMSTDOBJ].

<x>/Config/DeviceAppAuth/<x>/AppAuth/<x>/AAuthType

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This node specifies the authentication type. See details in [DMSTDOBJ].

<x>/Config/DeviceAppAuth/<x>/AppAuth/<x>/AAuthName

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This node specifies the authentication name. See details in [DMSTDOBJ].

<x>/Config/DeviceAppAuth/<x>/AppAuth/<x>/AAuthSecret

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node specifies the authentication secret. See details in [DMSTDOBJ].

<x>/Config/DeviceAppAuth/<x>/AppAuth/<x>/AAuthData

Status	Occurrence	Format	Min. Access Types
Required	One	bin	No Get

This node specifies the authentication data. See details in [DMSTDOBJ].

<x>/DevGroup

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This interior node contains information pertaining to all device groups that have been set up on the DM Gateway for the group management of devices that are subtending from the DM Gateway.

The DM Server uses the group identifier to fanout commands to multiple End Devices, via a DM Gateway operating in the Proxy Mode.

<x>/DevGroup/<x>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This placeholder node contains information pertaining to one device group.

<x>/DevGroup/<x>/GroupID

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get, No Replace

The value of this node specifies the device group identifier. This value MUST be unique within the Management Tree of the DM Gateway. The value of this node MUST be set by the DM Gateway.

The DM Gateway SHOULD follow some naming convention for device groups to ensure that the device group identifier does not clash with any End Device identifier.

<x>/DevGroup/<x>/GroupSize

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get, No Replace

The value of this leaf node is the number of devices within the device group. The value of this node MUST be set by the DM Gateway.

<x>/DevGroup/<x>/GroupType

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get, Replace

The value of this leaf node indicates the criteria for membership of this group. The value of this node MUST be one of the following:

Value	Semantics
0	Enumerated membership (default value)
1	Membership based on device type
2	Membership based on other conditions set by the DM Server

If the DM Server sets the value to be '0', then the DM Server MUST add individual End Devices under <x>/DevGroup/<x>/Members sub-tree.

If the DM Server sets the value to be '1' or '2', then the DM Gateway MUST add individual End Devices under <x>/DevGroup/<x>/Members sub-tree according to the conditions specified in <x>/DevGroup/<x>/Condition node.

When the <x>/DevGroup/<x>/Members sub-tree is not empty, the DM Server MUST NOT change the value of this node. Any attempt by the DM Server to change the value of this node when the device group is not empty SHALL be rejected by the DM Gateway, with a status code '403 forbidden'.

<x>/DevGroup/<x>/Members

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This interior node contains all of the members of the device group.

<x>/DevGroup/<x>/Members/<DevID>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	null	Get, Add, Delete (in the case of enumerated membership) Get, No Add, No Delete , No Replace (in the case of device type based membership or other criteria based membership)

The name of this placeholder leaf node is the identifier of the device that is the member of the device group. Any attempt by the DM Server to manipulate (Add, Delete or Replace) this node if the value of the related *GroupType* node is '1' (that is, membership based on device type) or '2' (that is, membership based on other conditions set by the DM Server) MUST be rejected by the DM Gateway, with a status code '403 forbidden'.

<x>/DevGroup/<x>/Condition

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node is the parent node for all of the information pertaining to a condition-based group.

<x>/DevGroup/<x>/Condition/DevTypeCriteria

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get, Replace

The value of this leaf node is the device type associated with the device group set by the DM Server. This node is mutually exclusive with the *<x>/DevGroup/<x>/Condition/OtherCriteria* node.

<x>/DevGroup/<x>/Condition/OtherCriteria

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get, Replace

The value of this leaf node is the condition expression that the DM Server wants the End Device to fulfill. The value of this node is set by the DM Server.

The expression is defined using the following ABNF syntax:

```
Expression = Condition *(( "|" / "&" ) Condition)

Condition = CondStr / "(" CondStr ")"

CondStr = URI ( "=" / ">" / "< " / "!=" ) Value

            ; Definition of URI is as per the TND spec

Value = ValueStr / "(" ValueStr ")" / \" ValueStr \"

ValueStr = 1*ValueChar

ValueChar = ALPHA / DIGIT / "+" / "_" / "." / " "
```

An example of the value of this node is:

```
(./A/B/Software1/VERSION=1.20 | ./A/B/Software1/VERSION<1.20)
& ./A/DevDetail/DevType=Smartphone
```

This node is mutually exclusive with the *<x>/DevGroup/<x>/Condition/DevTypeCriteria* node.

<x>/DevGroup/<x>/PopulateGroup

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	null	Get, Exec

This node is used with the Exec command by the DM Server to populate the group when the value of its sibling *GroupType* node is '1' or '2'. This node MUST NOT be present if the value of its sibling *GroupType* node is '0' (that is, *Enumerated membership*).

Once the DM Gateway receives the Exec command, it MUST add individual End Devices under *<x>/DevGroup/<x>/Members* sub-tree according to the conditions specified in the *<x>/DevGroup/<x>/Condition* node.

<x>/Privileges

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node groups information about the Target IDs (DeviceID or GroupID) with respective list of URIs along with corresponding list of Privilege Limitation rights (Privilege ACL).

<x>/Privileges/<x>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This placeholder node groups information about the handling of one particular type of Target ID (device or group of devices) by the DM Gateway.

<x>/Privileges/<x>/TargetID

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

The value of this leaf node specifies the GroupID or Device ID. The type of the identifier is determined by the TargetType node, which is present in the same sub-tree. The format of the Group ID would be as specified in the DevGroup sub-tree of this MO. And the format of the DeviceID would be as specified in Inventory MO.

<x>/Privileges/<x>/TargetType

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

The value of this leaf node indicates the type of the TargetID which is placed in the same sub-tree. The value of this node MUST be one of the following:

Value	Semantics
0	GroupID
1	DeviceID

The DM Gateway SHOULD follow some naming convention for device groups to ensure that the device group identifier does not clash with any End Device identifier.

<x>/Privileges/<x>/<x>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This node is a place holder for the URI and the corresponding PrivilegeACL of the TargetID.

<x>/Privileges/<x>/<x>/URI

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

The value of this leaf node specifies the URI (resource) of the end device for which the Privilege Access permissions are applicable. (Example: Enabling Camera).

<x>/Privileges/<x>/<x>/PrivilegeACL

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the permissions applied to allow Server management operations for the controlled resource of the respective target id in the DM Gateway.

In case of at least one Primary DM server account present on the device - to add or change rights for other servers , the gateway, receives a device management command to this node from primary server to add/modify an Privilege ACL for secondary server who doesn't have access right on a certain resource in a device or device group. (Refer section 8.3.4.2 for detail flows.) The value of this node SHOULD comply with ACL syntax defined in [DMTND].

<x>/Ext

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node is for vendor-specific extensions to the Gateway Config MO.

6.3 Fanout MO

6.3.1 MO Description

The Fanout MO resides in the Management Tree [DMTND] of the DM Gateway and maintains information regarding the handling of DM command fanout and response aggregation in the Proxy Mode or in the Adaptation Mode.

Figure 4 gives the pictorial description of the Fanout MO. The descriptions of the various nodes within this MO are provided below.

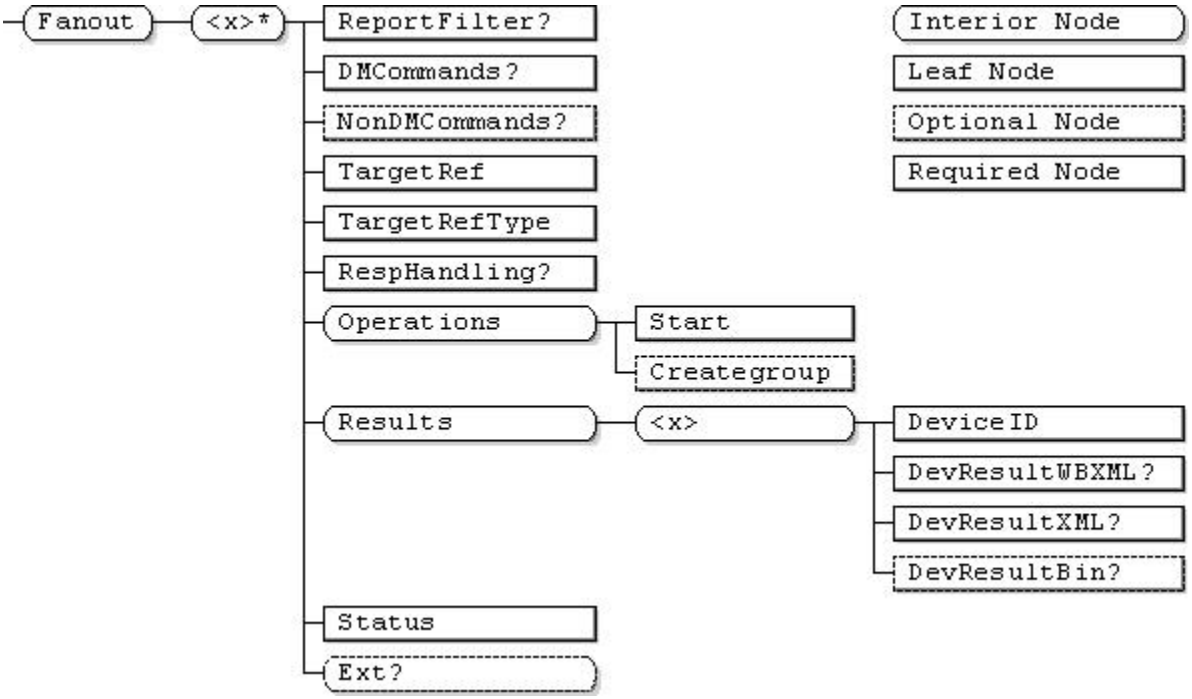

Figure 4: Fanout MO

Fanout

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is the root node for the Fanout MO.

The Management Object Identifier for the Fanout MO MUST be: “urn:oma:mo:oma-gwmo-fanout:1.0”.

Fanout/<x>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node is the placeholder for parameters regarding fanout operation for targeted End Device(s).

Fanout/<x>/ ReportFilter

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get, Replace

The value of this node specifies the filter condition for aggregation results; it can also be used to create the filtered group when 'creategroup' operation is used.

The expression for the Response filter condition is defined using the following ABNF syntax:

```
Expression = Condition *(("&" / "|") Condition)

Condition = "CmdID" ("="/ ">" / "<") CmdIDs "&" "(" StatusCondition ")"

StatusCondition = StatusVal *( ("&" / "|") StatusVal)

StatusVal = "StatusCode" ("=" / ">" / "<") StatusCodes

CmdIDs = 1*DIGIT

StatusCodes = 3*4DIGIT / "*"
```

This node specifies the combination of CmdID(s) and related status code(s) which indicates how the DM Gateway generates a Result Aggregation Alert based on the results from the End Devices. If it is present, the DM Gateway MUST send the filtered result within the Result Aggregation Alert based on the specific status code(s) for the specific CmdID(s) to the DM Server.

For example:

If the DM Server wants to indicate the expected format of results based on all status code for the 'Replace' command with CmdID 10001, then the DM Server can set the *Fanout/<x>/ReportFilter* node to "cmdid=10001&(statuscode>0 & statuscode < 999)" or "cmdid=10001&(statuscode= *)" .

If the DM Server wants to indicate the expected format of results based on a specific status code (say '200' or '202') for the 'Replace' command with CmdID 10001, then the DM Server can set the *Fanout/<x>/ReportFilter* node to "cmdid=10001&(statuscode=200 | statuscode=202)".

Fanout/<x>/DMCommands

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	xml	Get

This node specifies the management operations to be forwarded to all targeted End Devices specified by the TargetRef node when the 'Start'operation is executed by the DM Server. The value of this node MUST conform to the structure of the <SyncBody> element, as per the DM representation protocol [DMREPPRO].

This node is mutually exclusive with its sibling *NonDMCommands* node.

Fanout/<x>/NonDMCommands

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bin	Get

The value of this leaf node is the non-OMA DM message that will be forwarded to all targeted End Devices specified by the *'TargetRef'* node when *'Start'* operation is executed by the DM Server.

This leaf node is valid for non-OMA DM End Devices for which DM Gateway is operating in Adaptation Mode.

This node is mutually exclusive with its sibling *DMCommands* node.

Fanout/<x>/TargetRef

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get, Replace

The value of this leaf node specifies the intended target of the DM commands stored in the 'DMCommands' node. Depending upon the value of the 'TargetRefType' node, the DM Gateway will either issue the DM commands to the targeted End Device specified by the DeviceID node in the Device Inventory MO, or it will issue the DM commands to a device group specified by a GroupID node in the Gateway Config MO.

Fanout/<x>/TargetRefType

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get, Replace

The value of this leaf node indicates whether the value of *TargetRef* node is Device ID or Group ID. The value "0" indicates the TargetRef is the Group ID, and the value "1" indicates the TargetRef is the single Device ID.

Fanout/<x>/RespHandling

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

The value of this leaf node indicates how the response of the fanout command needs to be reported to the DM Server. The valid values for this node are as per the following table:

Value	Semantics
0	No report Upon completion of the fanout command, the DM Gateway MUST NOT send any alert to the DM Server. However, the command results are still available in the Results subtree and the DM Server can retrieve the complete results from there.
1	Completion Status report Upon completion of the fanout command, the DM Gateway MUST send the *Fanout Completion Status Alert*. This alert contains only the overall status of the command.
2	Aggregated Response report Upon completion of the fanout command, the DM Gateway MUST send the *Fanout Result Aggregation Alert*. This alert contains not only the overall status of the command but the aggregated response as well.

If this node is not present, the default value of 1 (that is, *Completion Status* report) MUST apply.

Fanout/<x>/Operations

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is the parent node for operations supported by the Fanout MO.

Fanout/<x>/Operations/Start

Status	Occurrence	Format	Min. Access Types
Required	One	null	Get, Exec

This node is the target node for the Exec command to start the command fanout operation. Invoking Exec on this node causes the DM Gateway to send the DM commands specified in the *<x>/FanOut/DMCommands* node to the target End Device(s).

Fanout/<x>/Operations/Creategroup

Status	Occurrence	Format	Min. Access Types
Optional	One	null	Get, Exec

This Optional node is used with the Exec command by the DM Server to request the DM Gateway to populate a new group based on the criteria specified in *Fanout/<x>/ReportFilter* node. After the DM Gateway performs the Exec command, the DM Gateway MUST populate the device group within the Gateway ConfigMO and send the URI of the new group to the DM Server.

Fanout/<x>/Results

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

The interior node contains the response from each End Device to which the DM command was fanned out.

Fanout/<x>/Results/<x>

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get, Delete

This placeholder node contains the response from one End Device.

Fanout/<x>/Results/<x>/DeviceID

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

The value of this leaf node specifies the identifier of the End Device.

Fanout/<x>/Results/<x>/DevResultWBXML

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bin	Get

The value of this leaf node is the <SyncBody> of the response message received by the DM Gateway from the End Device in wbxml format. The <SyncHdr> part of the response message MUST be removed by the DM Gateway.

This node is mutually exclusive with its sibling DevResultXML and DevResultBin node.

Fanout/<x>/Results/<x>/DevResultXML

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	xml	Get

The value of this leaf node is the <SyncBody> of the response message received by the DM Gateway from the End Device in xml format. The <SyncHdr> part of the response message MUST be removed by the DM Gateway.

This node is mutually exclusive with its sibling DevResultWBXML and DevResultBin node.

Fanout/<x>/Results/<x>/DevResultBin

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bin	Get

The value of this leaf node is the response for the fanout command, as per the native management protocol supported by the End Device.

This node is mutually exclusive with the *DevResultWBXML* and *DevResultXML* nodes.

Fanout/<x>/Status

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get, No Replace

The value of this leaf node indicates the status of the fanout command. The allowed values for this node are as per the following table:

Value Range	Semantics
0	Command not executed (default value)
1	Command under execution
2	Command completed successfully
3	Command completed with errors

Fanout/<x>/Ext

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node is for vendor-specific extensions to the Fanout MO.

6.4 Image Inventory MO

6.4.1 MO Description

The Image Inventory MO resides in the Management Tree [DMTND] of the DM Gateway and it maintains information regarding images (for example, Delivery Package for SCOMO) which can be retrieved by or delivered to End Device(s). This MO can also be utilized for efficiently delivering images to multiple End Devices combined with, but not limited to, fanout commands.

Figure 5 gives the pictorial description of the Image Inventory MO. The description of the related nodes within this MO is given below.

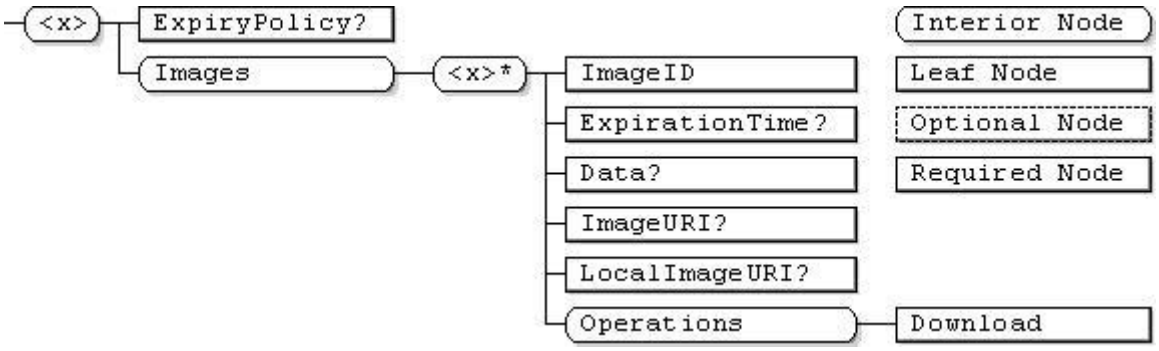

Figure 5: Image Inventory MO

<x>

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This place holder node is the root node for the Image Inventory MO. The parent node of this node defines the location of this MO in the DM Gateway's Management Tree.

The MOID for the Image Inventory MO MUST be: "urn:oma:mo:oma-gwmo-imageinventory:1.0".

<x>/ExpiryPolicy

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get, No Replace

This leaf node specifies the maximum amount of time, in minutes, for which the DM Gateway can locally cache any image. The purpose for this node is to let the DM Server know how long the local image URI can be referenced by the DM Server for delivering images to the End Devices. This value MUST be set by the DM Gateway.

Absence of this node implies that the DM Gateway is not resource constrained and it can store image(s) as long as the DM Gateway's policy permits.

<x>/Images

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is the parent node for all the data pertaining to locally stored images.

<x>/ Images/<x>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This placeholder node contains information pertaining to a specific image.

<x>/ Images/<x>/ImageID

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the identifier for a delivered image. The value for this leaf node MUST be assigned by the DM Gateway and it MUST be unique within Image Inventory MO.

<x>/ Images/<x>/ExpirationTime

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This leaf node specifies the expiration date and time for this image. After the expiration time elapses, the DM Gateway SHOULD remove this image as soon as practical. The representation of this node MUST follow the [YYYY]-[MM]-[DD] T[hh]:[mm]Z format, as defined by ISO 8601.

If the *<x>/ExpiryPolicy* node exists, the DM Server MUST set this value so that the overall time the DM Gateway stores this image is less than or equal to the value of the *<x>/ExpiryPolicy* node. In the case that the *<x>/ExpiryPolicy* is not present, the DM Server can set this value to any date and time in the future.

If this leaf node is not present, it means that there is no expiration for this image, and the DM Gateway can keep this image as long as the DM Gateway's policy permits. In this case, this image can be deleted with an explicit Delete command.

<x>/ Images/<x>/Data

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bin	No Get, Replace

This leaf node stores the actual binary data for an image. When using OMA DM for image delivery, the DM Server can upload the actual image data by sending the Replace command to this node. After completely receiving the image, the DM Gateway MUST properly set the *LocalImageURI* node. For this, the DM Gateway can store the image in the DM Gateway itself or in a local storage server. The DM Gateway MUST send the *Image Ready* Alert back to the DM Server after setting the *LocalImageURI* node. This node is mutually exclusive with its sibling *ImageURI* node.

<x>/ Images/<x>/ImageURI

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This leaf node specifies the downloadable URI of the image; this value MUST be set by the DM Server. This URI is used by the DM Gateway to download the image by using an alternate download mechanism (such as HTTP Get or Download over the Air [DLOTA]). The DM Gateway MUST support HTTP Get to download the image. This node is mutually exclusive with the sibling *Data* node.

<x>/ Images/<x>/LocalImageURI

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get, No Replace

This leaf node specifies the local URI of an image, and its value MUST be set by the DM Gateway after the image is downloaded either by using the alternative download mechanism (using the *ImageURI* node) or by using OMA DM (using the *Data* node). The downloaded image can reside in the DM Gateway or in the local storage server.

<x>/ Images/<x>/Operations

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is the parent node for operations supported by the Image Inventory MO.

<x>/ Images/<x>/Operations/Download

Status	Occurrence	Format	Min. Access Types
Required	One	null	Get, Exec

This node is used with Exec command to download the image identified by the *ImageURI* node. After completely downloading the image, the DM Gateway MUST properly set the *LocalImageURI* node, and MUST send the *Image Ready Alert* back to the DM Server to inform the completion of the download.

6.5 End Device Trigger MO

6.5.1 MO Description

The End Device Trigger MO resides in the Management Tree [DMSTDOBJ] of a DM Gateway that supports the Transparent Mode of operation. This MO accepts the Notification message [DMNOTI] from the DM Server and forwards it to the End Device(s).

Figure 6 shows a pictorial description of the Gateway Notification MO. The descriptions of the various nodes within this MO are provided below.

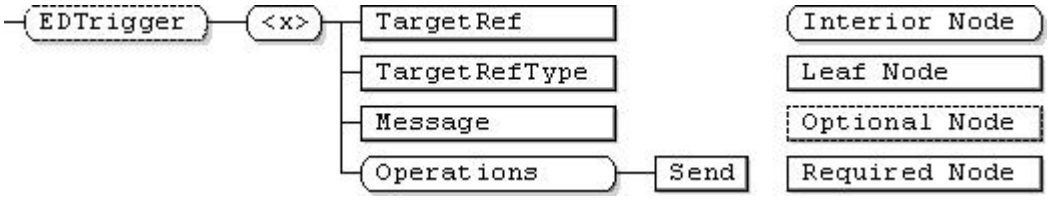

Figure 6: End Device Trigger MO

EDTrigger

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is the root node for the End Device Trigger MO. If the DM Gateway does not support DM Notification message forwarding using the End Device Trigger MO, this node MUST NOT be present.

The Management Object Identifier for the End Device Trigger MO MUST be: “urn:oma:mo:oma-gwmo-edtrigger:1.0”.

EDTrigger/<x>

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node is the placeholder for parameters regarding sending Notification message to the End Device(s).

EDTrigger/<x>/TargetRef

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get, Replace

The value of this leaf node references the intended target of the DM Notification message stored in the '*EDTrigger/Message*' node.

Depending upon the value of the '*EDTrigger/<x>/TargetRefType*' node, the DM Gateway will either send the DM Notification message to the targeted End Device specified by the *DeviceID* in Device Inventory MO, or it will send the DM Notification message to the targeted End Devices, specified by the *GroupID* in the Gateway Config MO.

EDTrigger/<x>/TargetRefType

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get, Replace

The value of this leaf node indicates whether the value of '*EDTrigger/<x>/TargetRef*' node is DeviceID or GroupID. The value "0" indicates the '*EDTrigger/<x>/TargetRef*' is a GroupID; the value "1" indicates the '*EDTrigger/<x>/TargetRef*' is a single DeviceID.

EDTrigger/<x>/Message

Status	Tree Occurrence	Format	Min. Access Types
Required	One	bin	Replace

The value of this node is the Notification message data as defined in [DMNOTI]. The DM Gateway will forward this content to the targeted End Device(s) pointed to by the '*EDTrigger/<x>/TargetRef*' node upon invocation of the Exec command on the *EDTrigger/<x>/Operations/Send* node.

EDTrigger/<x>/Operations

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This node is the parent node for operations within the Gateway Notification MO.

EDTrigger/<x>/Operations/Send

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Exec

This node is used with the Exec command to send the Notification message to the End Device specified in *EDTrigger/ <x>/TargetRef* node.

6.6 End Device Account Extension

6.6.1 MO Description

When an End Device queries a local Bootstrap Server and retrieves the bootstrap package, using the server(s) information provided in the bootstrap package, it is not readily apparent to the End Device whether a given server is a DM Server or a DM Gateway. To address this, the DM Account pertaining to DM GwMO MUST support the following additional node(s). Other nodes in DM Account are defined in [DMSTDOBJ].

<x>/Ext/oma_gwmo

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This placeholder node contains information about the DM Gateway specific behavior

<x>/Ext/oma_gwmo/ ServerType

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get

This leaf node specifies whether the server is a specialized DM Server operation (such as DM Gateway). If the ServerType node is set to true, then the server is DM Gateway. Else if it set to false, the server is the normal DM Server.

The default value of this node is false.

7. Alerts (Normative)

This section describes the various alerts that have been defined for realizing the DM Gateway [DMDICT] functionality.

7.1 Device Inventory Alerts

The device inventory alerts are issued by the DM Gateway when the DM Gateway becomes aware of a new End Device in the network or the DM Gateway becomes aware that a previously subtending End Device is no longer present in the network.

The device inventory alerts MUST carry an XML body that conforms to the schema specified in Appendix C.1. The alert content includes the following parameters:

- DeviceID: Identifier for the End Device.
- Address: Public routable address of the End Device (if applicable; included only in Transparent Mode).
- AddressType: Public routable address type of the End Device (if applicable; included only in Transparent Mode).
- Mode: Operation mode of the DM Gateway for this particular End Device (included only in Device Attach Alert).

The following sub-sections define the various types of device inventory alerts that are specified in this enabler.

7.1.1 Device Attach Alert

The *Device Attach Alert* is issued by the DM Gateway to the DM Server when the DM Gateway detects a new End Device. The mechanism by which a new End Device is detected is outside the scope of this specification.

The Device Attach Alert conforms to the Generic Alert [DMPRO] mechanism. The alert message includes the following data:

- <Meta>/<Type> element: Contains the media type of the alert content. The value MUST be the alert type identifier ‘urn:oma:at:oma-gwmo:deviceattached:1.1’.
- <Meta>/<Format> element: Contains the format of the alert content. The value MUST be ‘xml’.
- <Meta>/<Mark> element: Contains the importance level of the alert message. This element is optional.
- <Source>/<LocURI> element: Contains the source address of the MO. The value MUST be the address of the <x>/Inventory node.
- <Item>/<Data> element: Contains the device information of the attached devices in xml format as defined in Appendix C.1.

When the device information exists on the DM Server, the DM Server MUST update the device information according to the Device Attach Alert.

The following is an example of the Device Attach Alert message which reports four attached End Devices:

```
<Alert>
  <CmdID>2</CmdID>
  <Data>1226</Data>       <!-- Generic Alert -->
  <Item>
    <Source><LocURI>./Gateway/Inventory</LocURI></Source>
    <Meta>
      <Type xmlns='syncml:metinf'>
        urn:oma:at:oma-gwmo:deviceattached:1.1
      </Type>
      <Format xmlns='syncml:metinf'>xml</Format>
      <Mark xmlns='syncml:metinf'>informational</Mark>   <!-- Optional -->
```

```
    </Meta>
    <Data>
      <![CDATA[
        <DeviceInventory>
          <Device>
            <DeviceID>device123</DeviceID>
            <Mode>1</Mode>
          </Device>
          <Device>
            <DeviceID>device456</DeviceID>
            <Mode>1</Mode>
          </Device>
          <Device>
            <DeviceID>device789</DeviceID>
            <Mode>1</Mode>
          </Device>
          <Device>
            <DeviceID>device011</DeviceID>
            <Mode>2</Mode>
          </Device>
        </DeviceInventory>
       ]]>
     </Data>
   </Item>
</Alert>
```

7.1.2 Device Detach Alert

The Device Detach Alert is issued by the DM Gateway to the DM Server when an existing End Device is detached from the DM Gateway. The mechanism by which detachment of an existing End Device is detected is outside the scope of this specification.

The Device Detach Alert conforms to the Generic Alert [DMPRO] mechanism. The alert message includes the following data:

- <Meta>/<Type> element: Contains the media type of the alert content. The value MUST be the alert type identifier ‘urn:oma:at:oma-gwmo:devicedetached:1.0’.
- <Meta>/<Format> element: Contains the format of the alert content. The value MUST be ‘xml’.
- <Meta>/<Mark> element: Contains the importance level of the alert message. This element is optional.
- <Source>/<LocURI> element: Contains the source address of the MO. The value MUST be the address of the <x>/Inventory node.
- <Item>/<Data> element: Contains the device information of the detached devices in xml format as defined in Appendix C.1.

The following is an example of the Device Detach Alert message which reports two detached End Devices:

```
<Alert>
   <CmdID>2</CmdID>
   <Data>1226</Data>      <!-- Generic Alert -->
   <Item>
      <Source><LocURI>./Gateway/Inventory</LocURI></Source>
      <Meta>
         <Type xmlns='syncml:metinf'>
```

```
            urn:oma:at:oma-gwmo:devicedetached:1.0
          </Type>
          <Format xmlns='syncml:metinf'>xml</Format>
          <Mark xmlns='syncml:metinf'>informational</Mark>   <!-- Optional -->
      </Meta>
      <Data>
      <![CDATA[
        <DeviceInventory>
          <Device>
            <DeviceID>device123</DeviceID>
          </Device>
          <Device>
            <DeviceID>device012</DeviceID>
          </Device>
       ]]>
     </Data>
   </Item>
</Alert>
```

7.2 Command Fanout Alerts

The command fanout alerts are issued by the DM Gateway to the DM Server in response to a command fanout operation. The following sub-sections define the various types of command fanout alerts that are specified in this enabler.

7.2.1 Fanout Result Aggregation Alert

The Fanout Result Aggregation Alert is used by the DM Gateway to include the aggregated response of the fanout command from multiple End Devices. Prior to invoking a fanout command, the DM Server can specify that it wants the DM Gateway to issue a Fanout Result Aggregation Alert upon completion of the fanout command.

The Fanout Result Aggregation Alert conforms to the Generic Alert [DMPRO] mechanism. The alert message includes the following data:

- <Meta>/<Type> element: Contains the media type of the alert content. The value MUST be the alert type identifier 'urn:oma:at:oma-gwmo: resultaggregation 1.0'.
- <Meta>/<Format> element: Contains the format of the alert content. The value MUST be 'xml'.
- <Meta>/<Mark> element: Contains the importance level of the alert message. This element is optional.
- <Source>/<LocURI> element: Contains the source address of the MO. The value MUST be the URI of root of the Result sub-tree where thefanout command result is stored.
- <Item>/<Data> element: Contains the aggregated result in xml format, as defined in Appendix C.2.

The following is an example of the Fanout Result Aggregation Alert message which reports results for two End Devices:

```
<Alert>
  <CmdID>157</CmdID>
  <Data>1226</Data>       <!-- Generic Alert -->
  <Item>
    <Source><LocURI>./Gateway/Fanoutobject1/Results</LocURI></Source>
    <Meta>
      <Type xmlns='syncml:metinf'>
        urn:oma:at:oma-gwmo:resultaggregation:1.0
      </Type>
```

```
      <Format xmlns='syncml:metinf'>xml</Format>
      <Mark xmlns='syncml:metinf'>informational</Mark>
    </Meta>
    <Data>
      <![CDATA[
          <Node>
  <!— Result of Device123 -->
           <DeviceID>device123</DeviceID>
           <DevResult>                              <!— This Result contains the
whole SyncBody -->
              <SyncBody>
                <Status>
                  <MsgRef>1</MsgRef>
                  <CmdID>2</CmdID>
                  <CmdRef>1</CmdRef>
                  <Cmd>Sequence</Cmd><!-- Sequence executed correctly -->
                  <Data>200</Data>
                </Status>
                <Status>
                  <MsgRef>1</MsgRef>
                  <CmdRef>4</CmdRef>
                  <CmdID>4</CmdID>
                  <Cmd>Replace</Cmd>
                  <TargetRef>./xxxx/xxxxxx/xxxx</TargetRef>     <!-- OK, data
                  changed -->
                  <Data>200</Data>
                </Status>
                <Final/>
              </SyncBody>
           </DevResult>
         </Node>
         <Node>
  <!-- Result of Device456 -->
           <DeviceID>device456</DeviceID>
           <DevResult>                <!— This Result contains the whole
SyncBody -->
             <SyncBody>
               <Status>
                 <MsgRef>1</MsgRef>
                 <CmdID>1</CmdID>
                 <CmdRef>0</CmdRef>
                 <Cmd>SyncHdr</Cmd>   <!-- SyncHdr accepted, Server will ignore
                 this part-->
                 <Data>200</Data>
               </Status>
               <Status>
                 <MsgRef>1</MsgRef>
                 <CmdRef>4</CmdRef>
                 <CmdID>4</CmdID>
                 <Cmd>Replace</Cmd>
                 <TargetRef>./xxxx/xxxxxx/xxxx</TargetRef>
                 <Data>403</Data>                                  <!--No!
                 Forbidden -->
               </Status>
               <Final/>
             </SyncBody>
           </DevResult>
         </Node>
```

```
       ]]>
     </Data>
   </Item>
</Alert>
```

7.2.2 Fanout Completion Status Alert

The Fanout Completion Status Alert is used by the DM Gateway to indicate the completion of the fanout command.

The Fanout Completion Status Alert conforms to the Generic Alert [DMPRO] mechanism. The alert message includes the following data:

- <Meta>/<Type> element: Contains the media type of the alert content. The value MUST be the alert type identifier 'urn:oma:at:oma-gwmo:completionstatus 1.0'.
- <Meta>/<Format> element: Contains the format of the alert content. The value MUST be 'xml'.
- <Meta>/<Mark> element: Contains the importance level of the alert message. This element is optional.
- <Source>/<LocURI> element: Contains the source address of the MO. The value MUST be the URI of the root of the Result sub-tree where the fanout command result is stored.

The following is an example of the Fanout Completion Status Alert message:

```
The following is an example of the Fanout Completion Status Alert message:
<Alert>
  <CmdID>257</CmdID>
  <Data>1226</Data>       <!-- Generic Alert -->
  <Item>
    <Source><LocURI>./Gateway/Fanoutobject1/Results</LocURI></Source>
    <Meta>
      <Type xmlns='syncml:metinf'>
        urn:oma:at:oma-gwmo:completionstatus:1.0
      </Type>
      <Format xmlns='syncml:metinf'>xml</Format>
      <Mark xmlns='syncml:metinf'>informational</Mark>
    </Meta>
   </Item>
</Alert>
```

7.3 Bootstrapped DMS List Alert

The *Bootstrapped DMS List* Generic Alert is issued by the End Device to the DM Gateway under the following conditions:

- the End Device is bootstrapped to a new DM server
- the End Device is unbootstrapped from an existing DM Server

The Alert Type Identifier for this alert MUST be urn:oma:at:oma-gwmo:BootstrappedDMSList:1.0.

This alert conforms to the Generic Alert structure, as defined in [DMPRO]. It lists all the DM Servers to which the End Device is currently bootstrapped.

An example of this alert is shown below.

```
<Alert>
  <CmdID>2</CmdID>
  <Data>1226</Data>               <!-- Generic Alert -->
  <Item>
    <Source><LocURI>mac:01-ab-34-ef-69-0c</LocURI></Source>
    <Target><LocURI>192.168.4.1</LocURI></Target>
    <Meta>
      <Type xmlns="syncml:metinf">
        urn:oma:gwmo: BootstrappedDMSList:1.0
      </Type>
       <Format xmlns="syncml:metinf">text/plain</Format>
    </Meta>
    <Data>
<![CDATA[
  <ServerList>
    <ServerInfo>
      <ServerID>ServerA</ServerID>
      <BootstrapURL>http://servera.com/bootstrap</BootstrapURL>
    </ServerInfo>
    <ServerInfo>
      <ServerID>ServerB</ServerID>
      <BootstrapURL>http://serverb.com/bootstrap</BootstrapURL>
    </ServerInfo>
  </ServerList>
-- ]]>
    </Data>
  </Item>
</Alert>
```

The XML Schema for this alert is provided in AppendixC.3.

7.4 Associated Gateway Alert

The *Associated Gateway* Generic Alert is issued by the End Device to the DM Server. It contains the publicly routable address of the DM Gateway.

The Alert Type Identifier for this alert MUST be urn:oma:at:oma-gwmo:AssociatedGateway:1.0.

This alert conforms to the Generic Alert structure, as defined in [DMPRO].

An example of the *Associated Gateway Generic Alert* is shown below.

```
<Alert>
  <CmdID>9</CmdID>
  <Data>1226</Data>           <!-- Generic Alert -->
  <Item>
    <Source><LocURI>mac:01-ab-34-ef-69-0c</LocURI></Source>
    <Target><LocURI>http://www.telco_operator.com/mgmt-
server</LocURI></Target>
    <Meta>
      <Type xmlns="syncml:metinf">
        urn:oma:gwmo: AssociatedGateway:1.0
      </Type>
      <Format xmlns="syncml:metinf">text/plain</Format>
    </Meta>
    <Data>
<![CDATA[
  <GwAddress>2002:0:0:0:0:0:9da6:e219</GwAddress>
-- ]]>
    </Data>
```

```
  </Item>
</Alert>
```

The XML Schema for this alert is provided in AppendixC.4.

7.5 Image Ready Alert

The Image Ready Alert is issued by the DM Gateway to the DM Server upon the successful alternative download or direct delivery via OMA DM. This alert indicates to the DM Server the location where the image has been stored locally. This enables the DM Server to include the local URI of the image in the DM commands that target the End Devices, for efficient distribution of the image. This Image Ready Alert MUST be sent in both cases; the alternative download mechanism, and the OMA DM direct delivery.

On receiving the Image Ready Alert, the DM Server can use the <x>/Images/<x>/LocalImageURI for further image distributions.

The Image Ready Alert conforms to the Generic Alert [DMPRO] mechanism. The alert message includes the following data:

- <Meta>/<Type> element: Contains the media type of the alert content. The value MUST be the alert type identifier ‘urn:oma:at:oma-gwmo:imageready:1.0’.
- <Meta>/<Format> element: Contains the format of the alert content. The value MUST be ‘xml’.
- <Meta>/<Mark> element: Contains the importance level of the alert message. This element is optional.
- <Source>/<LocURI> element: Contains the source address of the MO. The value MUST be the address of the <x>/Images/<x>/ImageURI node in case of using the alternative download or the address of the <x>/Images/<x>/Data node in case of using OMA DM direct delivery.
- <Target>/<LocURI> element: Contains the target address of the MO. The value MUST be the value of the <x>/Images/<x>/LocalImageURI node.

The following is an example of the Image Ready Alert message:

```
<Alert>
  <CmdID>2</CmdID>
  <Data>1226</Data>       <!-- Generic Alert -->
  <Item>
    <Source><LocURI>./GW/ImgInvMO/Images/FW1.2/ImageURI</LocURI></Source>
    <Target><LocURI>
http://fe80::202:b3ff:fe1e:8329/scomo/packageID123</LocURI></Target>
    <Meta>
      <Type xmlns=’syncml:metinf’>
        urn:oma:at:oma-gwmo:imageready:1.0
      </Type>
      <Format xmlns=’syncml:metinf’>xml</Format>
      <Mark xmlns=’syncml:metinf’>informational</Mark>   <!-- Optional -->
    </Meta>
   </Item>
</Alert>
```

8. DM Gateway Functionality

This section describes the DM Gateway [DMDICT] functionality in terms of the MOs that are defined in section 6 and the alerts that are defined in section 7.

8.1 General Management Flow

This section describes the general management flow for the remote management of End Devices, via the DM Gateway. Different steps in the general management flow are defined in the following sub-sections.

8.1.1 Inventory Update Flow

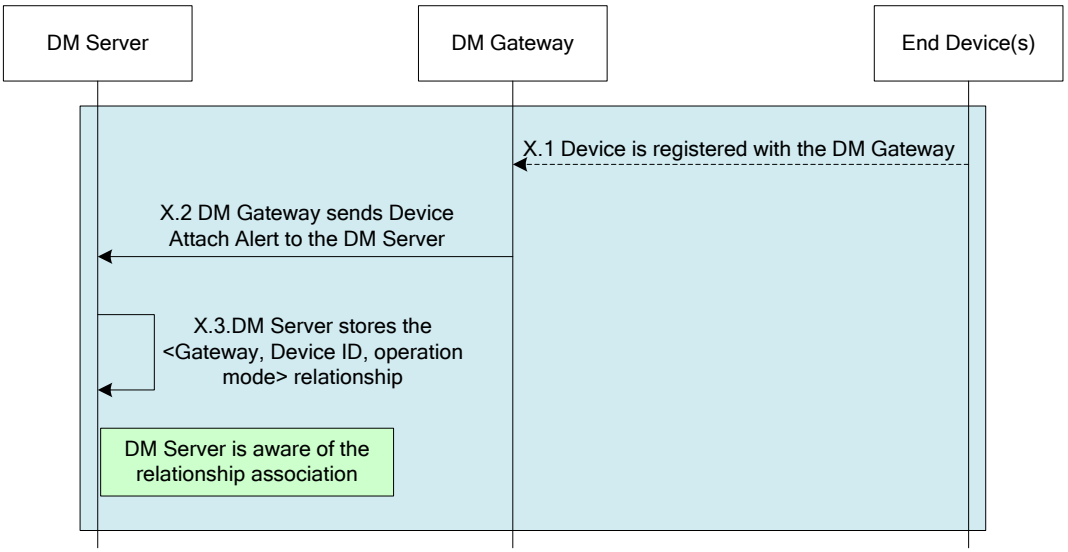

Figure 7: Inventory Update Flow

Step X.1: The End Device is detected by the DM Gateway.

Step X.2: The DM Gateway updates its Inventory MO and sends Device Attach Alert to the DM Server, which includes the Gateway address, Device ID/Address, and Operation mode.

Step X.3: The DM Server stores the <Gateway address, Device ID/Address, Operation mode> relationship. This is used to guarantee the DM Notification for the Device can be sent to the correct DM Gateway or Devices. At this stage, the DM Server becomes aware of the End Device.

8.1.2 Inventory Update Flow for Hierarchical Architecture

This flow only applies if the Hierarchical Architecture of DM Gateways is supported.

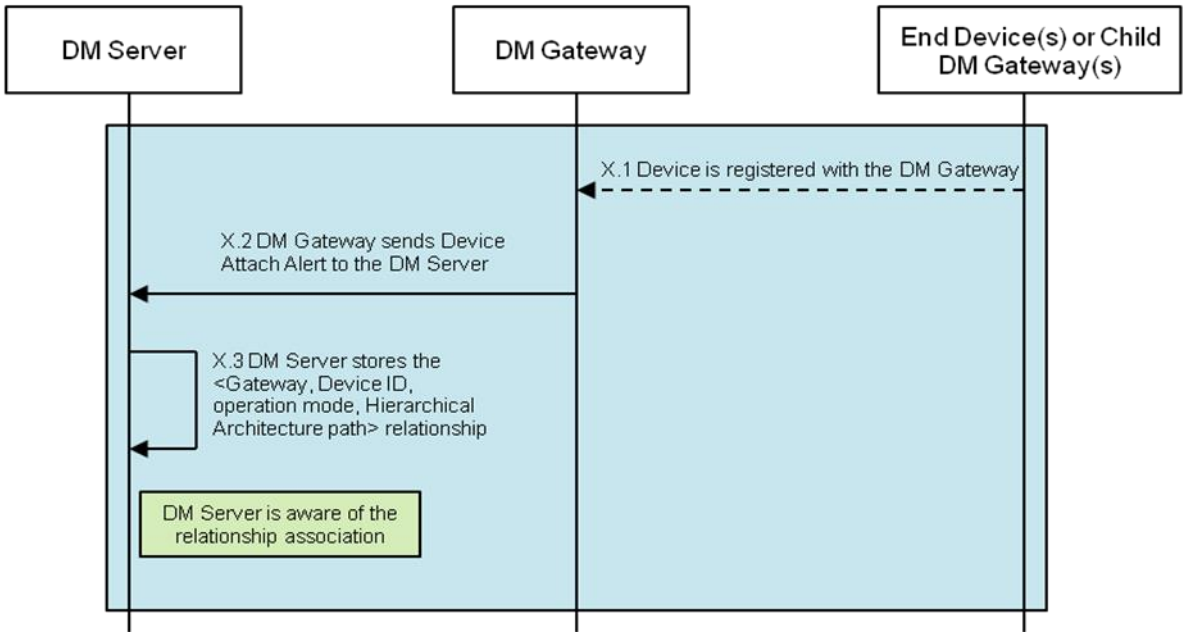

Figure 8: Inventory Update Flow for Hierarchical Architecture

Step X.1: The End Device or Child DM Gateway is detected by the DM Gateway.

Step X.2: DM Gateway updates its Inventory MO and it sends the Device Attach Alert to the DM Server via one or more intermediate DM Gateways in a tree branch. Only the DM Gateway attached to the detected device has its Inventory MO updated, the other intermediate DM Gateways do not have their Inventories MOs affected.

The Device Attach Alert SHALL include the Gateway address, Device ID/Address and Operation mode. Additionally, it MAY include the optional Hierarchical Architecture path, which contains the sequence of DM Gateways in the tree branch starting from the DM Gateway directly connected to the DM Server and ending with the DM Gateway connected to the target device.

If an End Device has been attached to the DM Gateway directly connected to DM Server, the Hierarchical Architecture path field is not required and field SHOULD be omitted.

Compatibility Note: This addresses the backward compatibility with GwMO Server V1.0 in a flat architecture (one level of Gateways), according to 8.1.1.

Step X.3: The DM Server stores the <Gateway address, Device ID/Address, Operation mode, Hierarchical Architecture path> relationship. DM Server SHALL be able to handle the optional Hierarchical Architecture path field as follows:

- if present, DM Server SHALL store the path and will include it in DM Notifications/Commands to be sent from the DM Server to the target End Devices or Child DM Gateways. The optional Hierarchical Architecture path field will be used by the intermediate DM Gateways to determine the "next hop while routing the DM Notification/Command to the DM Gateway associated to the target device.
- if absent, DM Notification/Command is to be sent to an End Device or child DM Gateway attached to the DM Gateway which directly communicates with the DM Server.

  Compatibility Note: This addresses the backward compatibility with GwMO ComponentV1.0 and GwMO Client V1.0 in a flat architecture (one level of Gateways), according to 8.1.1.

At this stage, the DM Server becomes aware of the End Device or Child DM Gateway and it is able to manage it through DM Notifications/Commands within the Hierarchical Architecture tree.

8.1.3 Synchronous Management in Proxy/Adaptation Mode

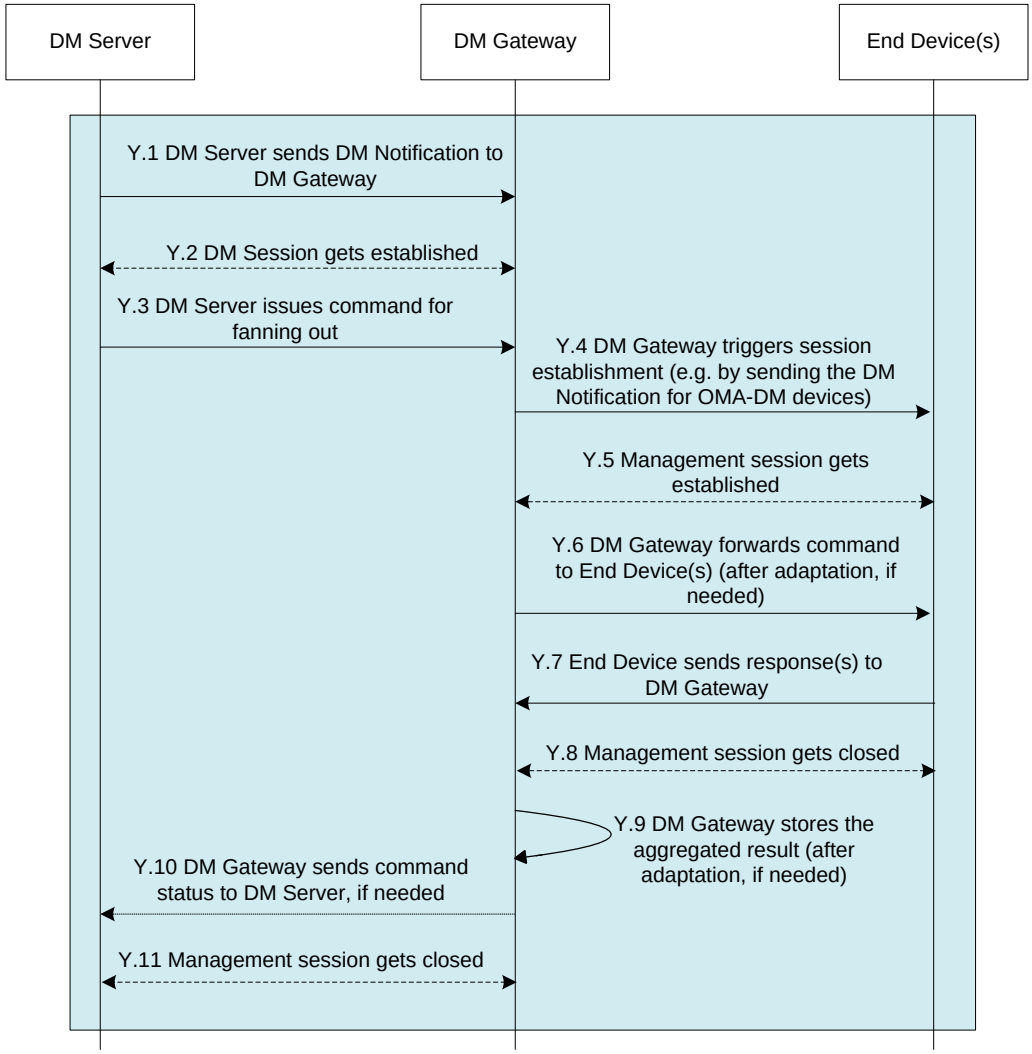

Figure 9: Synchronous Management in Proxy/Adaptation Mode

Step Y.1: The DM Server gets the address of the associated DM Gateway by querying the stored relationship. The DM Server sends the DM Notification to the DM Gateway.

Step Y.2: The DM session gets established between the DM Server and the DM Gateway.

Step Y.3: The DM Server instantiates the Fanout MO on the DM Gateway, indicating the target End Device(s). The DM Server subsequently invokes fanout execution.

Step Y.4: For each OMA DM target End Device, the DM Gateway generates its own DM Notification to the End Device. For each non-OMA DM target End Device, the mechanism used for triggering the End Device is outside the scope of this specification.

Step Y.5: An OMA DM management session is established between each OMA DM End Device and the DM Gateway. The protocol used for establishing a management session between non-OMA End Devices and the DM Gateway is outside the scope of this specification.

Step Y.6: The DM Gateway forwards the DM commands to the End Device(s) after adaptation, if needed.

Step Y.7: The End Device(s) send responses to the DM Gateway.

Step Y.8: The management session between the DM Gateway and the End Device(s) is closed.

Step Y.9: The DM Gateway stores the aggregated result after adaptation, if needed.

Step Y.10: If the DM Server chooses to be notified, the DM Gateway will send the *Fanout Result Aggregation* alert or the *Fanout Completion Status* alert to the DM Server, depending upon the response handling selection made by the DM Server in the Fanout MO.

Step Y.11: The OMA DM session between the DM Gateway and the DM Server is closed.

8.1.4 Asynchronous Management in Proxy/Adaptation Mode

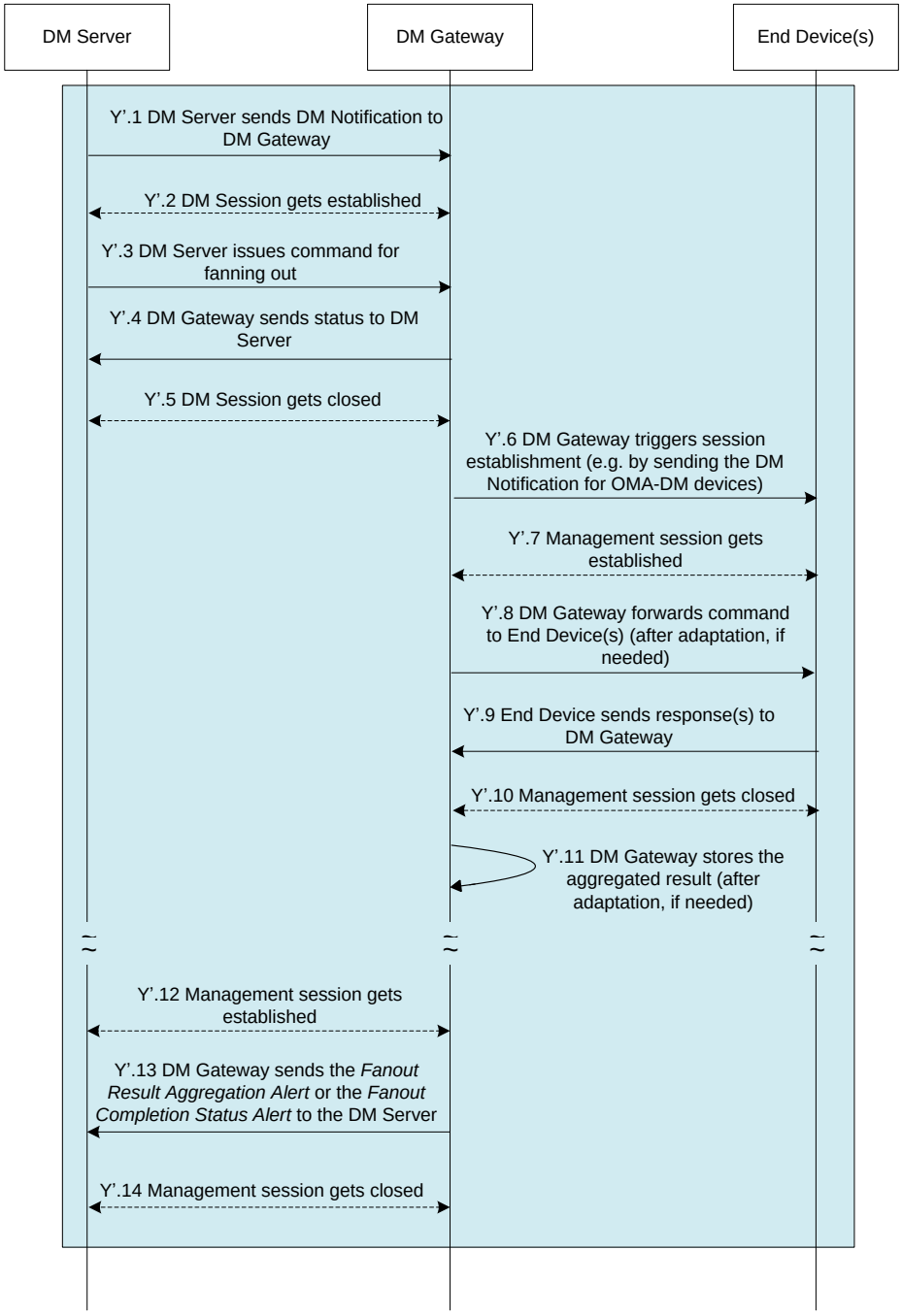

Figure 10: Asynchronous Management in Proxy/Adaptation Mode

Step Y'.1– Step Y'.3: These steps are identical to steps Y.1 through Y.3 in the 'Synchronous management flow in proxy/adaptation mode' case.

Step Y'.4: The DM Gateway sends the status to the DM Server indicating that the DM commands will be performed asynchronously.

Step Y'.5: The DM session gets closed.

Step Y'.6 – Step Y'.11: These steps are identical to steps Y.4 through Y.9 in the 'Synchronous management flow in proxy/adaptation mode'case.

Step Y'.12: OMA DM session gets established between the DM Server and the DM Gateway.

Step Y'.13: The DM Gateway sends the *Fanout Result Aggregation* alert or the *Fanout Completion Status* alert to the DM Server, depending upon the response handling selection made by the DM Server in the Fanout MO.

Step Y'.14: The OMA DM session between the DM Server and the DM Gateway gets closed.

8.1.5 Management in Transparent Mode

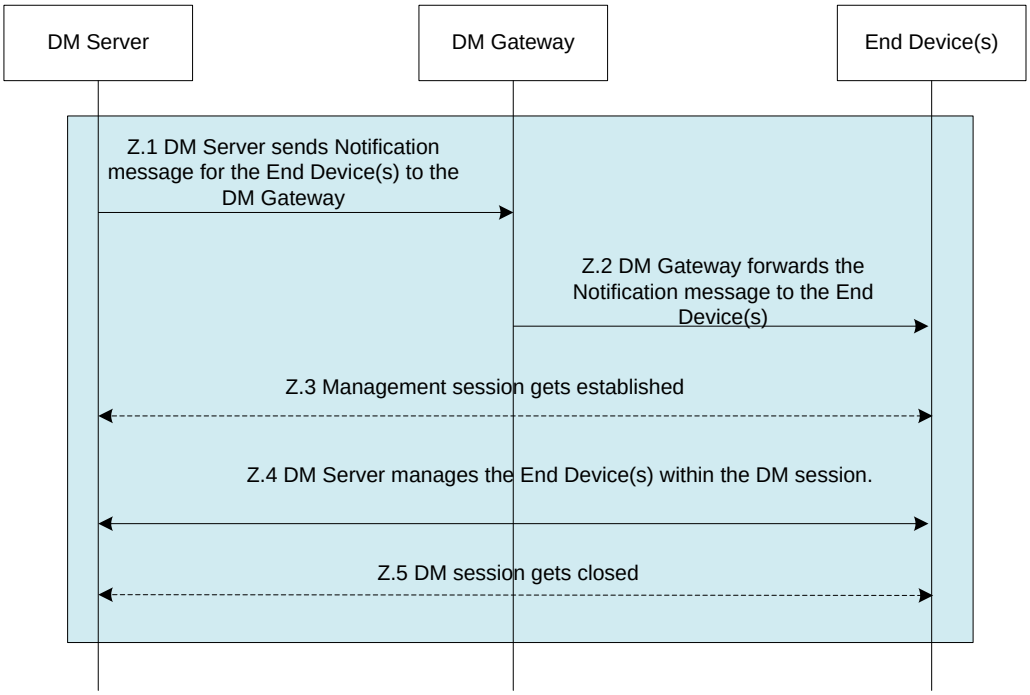

Figure 11: Management Flow in Transparent Mode

Step Z.1: The DM Server sends a DM Notification message for the End Device(s) to the DM Gateway. The Notification message indicates the target End Device(s) in a Push Extension Header field (section 8.2.1.1).

Step Z.2: The DM Gateway forwards the DM Notification message to the End Device(s).

Step Z.3: The End Device establishes a normal OMA DM session with DM Server.

Step Z.4: The DM Server manages the End Device within the context of the OMA DM session.

Step Z.5: The OMA DM session between the DM Server and the End Device gets closed.

8.2 Transparent Mode Operation

The Transparent Mode can be used by the DM Server to send the DM Notification message [DMNOTI] to a single End Device or to multiple End Devices, via the DM Gateway. The latter mode of operation is referred to in this specification as the *notification fanout* feature. For using the notification fanout feature, it is RECOMMENDED that the same notification credentials [DMSTDOBJ] be shared across multiple target End Devices and for the Notification message digest to be generated using these credentials.

This enabler defines two approaches for realizing the Transparent Mode. A DM Gateway MAY support one or both the approaches. These approaches are defined in the following sub-sections.

8.2.1 Push Header Extension Approach

This approach relies on a specially formatted OMA Push message [PushMsg], which contains the GwMO Extension Header, carrying OMA-DM Package #0 in its payload.

8.2.1.1 GwMO Package #0 Push Message Header Format

The GwMO Push Extension Header follows the same general structure as the other Push message headers, as described in [PushMsg]. This header is registered with the OMNA Push Message Header Code Registry under the following name:

- X-Oma-GwMO

The header structure is described by the following ABNF [RFC5234] syntax:

```
X-Oma-GwMO = "X-Oma-GwMO" ":" Address

Address = Target [ "&" Address ]

Target =  1*TargetChar

                  ; Target can be a device identifier or

                  ; a device group identifier

TargetChar = ALPHA / DIGIT / "." / "-" / "_" / " "
```

8.2.1.2 GwMO Package #0 Push Message Header Processing

To have a DM Notification message delivered to End Devices that do not have a publicly routable address, a DM Server MUST send the DM Notification message to the associated DM Gateway. The DM Notification message MUST include the *X-Oma-GwMO* extension header to specify the device identifier or a device group identifier. Upon receiving this message, the DM Gateway MUST process the X-Oma-GwMO header to find the target End Device(s) and then forward the DM Notification message to the specified End Device(s), without the X-Oma-GwMO header. The DM Gateway MUST NOT report any errors back to the DM Server.

The DM Server MAY also use this mechanism to send a DM Notification message to End Devices that have a publicly routable address.

The target referred to in the X-Oma-GwMO extension header can be:

- an End Device identifier
- a device group identifier

- an enumerated collection of End Device identifiers and/or device group identifiers, delimited by the "&" character

Upon receiving a DM Notification message, that contains the "X-Oma-GwMO" header, the DM Gateway first checks the Device Inventory MO to see if it recognizes the End Device identifier(s). It then checks the Gateway Config MO to see if it recognizes the device group identifier(s). The DM Gateway fans out the Notification to all of the target End Devices that it can resolve. Any unresolved Notification message targets are silently ignored.

Figure 12 shows an illustrative network consisting of a DM Server, a DM Gateway and three End Devices whose public identifiers are "Tab1", "Tab2", and "Tab3". The connectivity between all network elements in the diagram may be wired or wireless, and all network elements may be fixed or mobile.

.

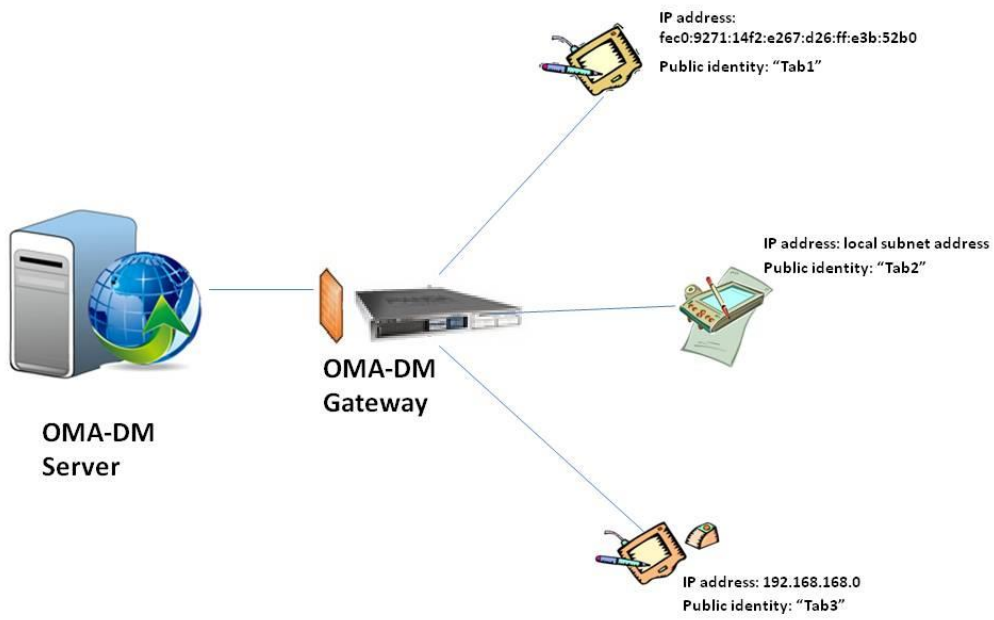

Figure 12: Illustrative Network of DM Server, DM Gateway and End Devices.

A DM Server that wants to send a DM Notification message to "Tab1" and "Tab3" will send a special Package 0 message to the DM Gateway. The message will contain the "X-Oma-GwMO" header, with the value set to "Tab1&Tab3".

Upon receiving this message, the DM Gateway establishes that the message is not destined for itself. It then checks its *Device Inventory* to see if it recognizes the target devices. In this case, the DM Gateway recognizes the devices and it forwards the Package 0 message to both the devices.

If the DM Gateway receives a message with the value of the "X-Oma-GwMO" header set to "Tab4", the message will be silently discarded.

8.2.2 End Device Trigger MO Approach

If the DM Gateway supports the End Device Trigger MO, the DM Server MAY use it for forwarding a DM Notification message to the End Device(s).

The Server will specify the targeted End Devices on ‘EDTrigger/<x>/TargetRef’ node and will set the value of ‘EDTrigger/<x>/TargetRefType’ according to the type of ‘EDTrigger/<x>/TargetRef’ node value. The actual DM Notification data is stored as the value of ‘EDTrigger/<x>/Message’ node.

When the Exec operation on the node ‘EDTrigger/<x>/Operations/Send’ is performed, the DM Gateway sends the content of ‘EDTrigger/<x>/Message’ to the targeted End Device(s). If the ‘EDTrigger/<x>/TargetRef’ value is not correctly set, the Gateway will return the error code “400: Bad Request” on the Exec operations.

8.2.2.1 MO Cleanup on the DM Gateway

In order to avoid proliferation of End Device Trigger MO instances on the DM Gateway, the DM Server SHOULD delete MO instances from the Management Tree when they are no longer in use.

8.3 Proxy Mode Operation

8.3.1 DM Command Fanout

The Fanout MO enables the DM Server to send the same DM commands to a group of End Devices. The DM Gateway MUST collect the response from each End Device within this group and make it available for retrieval by the DM Server from the Results subtree of the Fanout MO. Additionally, the DM Gateway MAY send the *Fanout Result Aggregation* alert or the *Fanout Completion Status* alert to the DM Server, depending upon the response handling selection made by the DM Server in the Fanout MO.

The following diagram describes the Fanout flow in case the DM Gateway is configured to send the Fanout Result Aggregation Alert to the DM Server:

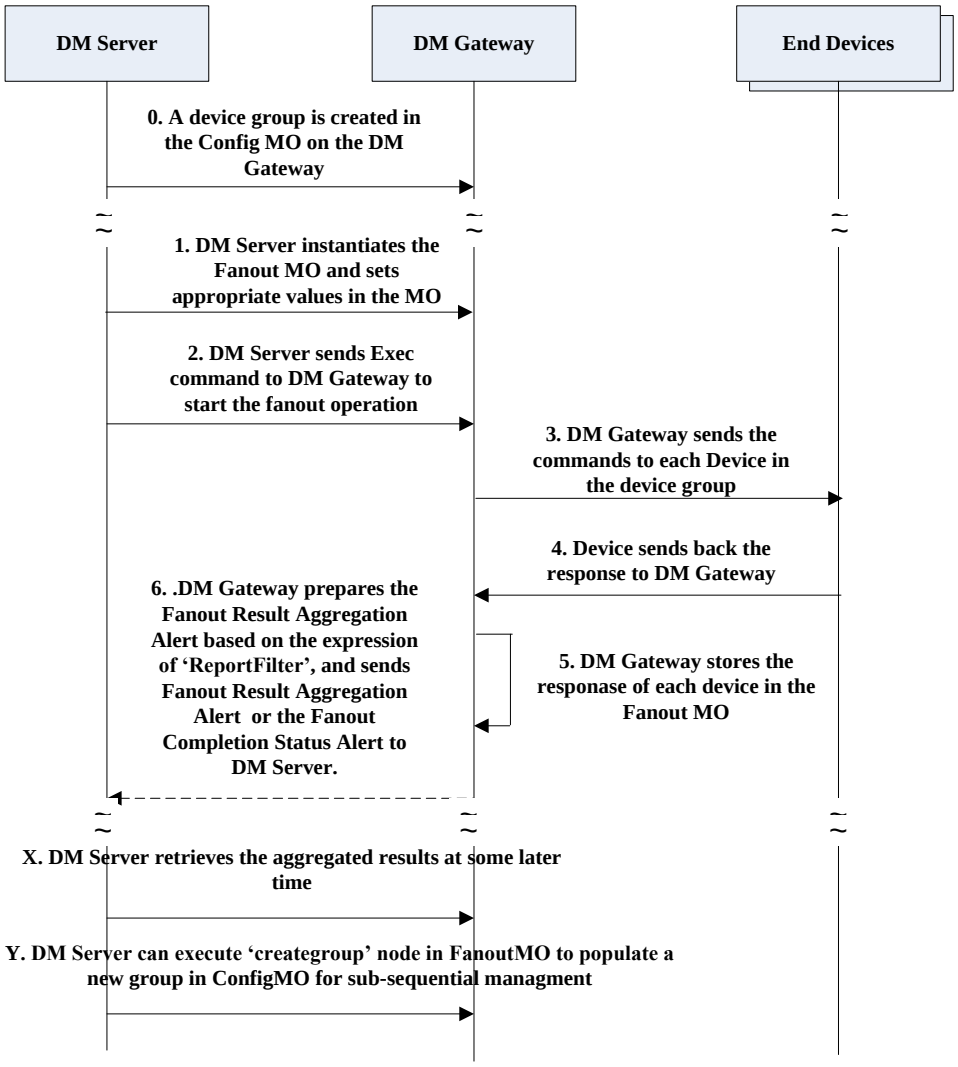

Figure 13: DM Command Fanout

Step 0: The pre-condition of Fanout operation is that a device group has been created in the Gateway Config MO by the DM Gateway.

Step 1: The DM Server instantiates the Fanout MO and sets the following nodes in the Fanout MO with the appropriate values:

FanOut/<x>/DMCommands

FanOut/<x>/TargetGroupRef

If the DM Server would like to obtain the specific status code(s) for a specific CmdID, then it sets this node with the appropriate value:

FanOut/<x>/ReportFilter

An example to set up *ReportFilter* node: *CmdID=100 & StatusCode=404.*

Step 2: The DM Server sends the Exec command to the DM Gateway to trigger it to fan-out DM Commands set in *FanOut/<x>/DMCommands node* to the targeted End Device(s) specified in the*<x>/FanOut /TargetRef node*.

For example:

```
<Exec>
   <CmdID>3</CmdID>
    <Item>
       <Target>
       <LocURI>./SCM/FanOut/Fanoutinstance01/Operation/Start</LocURI>
       </Target>
    </Item>
</Exec>
```

Step 3: The DM Gateway establishes a DM Session with each End Device specified in the *FanOut /<x>/TargetRef node* and forwards the DM Commands set in *FanOut /<x>/DMCommands* node to the targeted End Device(s).

Step 4: The End Device receives the DM Commands from the DM Gateway and returns the results to the DM Gateway after execution.

Step 5: The DM Gateway stores the response from each End Device in the Fanout MO. For each End Device, the DM Gateway creates an interior child node under the *FanOut /<x>/Results* node. This node contains the following leaf child nodes:

- *<x>/Results/<x>/DeviceID:* The value of this leaf node is the identifier of the End Device
- *<x>/Results/<x>/DevResultXML* (if the command result is in the SyncML format) or *<x>/Results/<x>/DevResultWBXML* (if the command result is in the WBXML format): The value of this leaf node is the command result returned by the End Device

Step 6: After the DM Gateway collects all responses, it will check the value of 'ReportFilter' node and prepare the Fanout Result Aggregation Alert:

If the node 'ReportFilter' was specified, then the Fanout Result Aggregation Alert will contain the results from End Device(s) which return the specified cmdid and its related status code;

If the node 'ReportFilter' was not specified, then the Fanout Result Aggregation Alert will contain all the results from End Device(s).

After the preparation of the Fanout Result Aggregation Alert, the DM Gateway MAY send the *Fanout Result Aggregation* alert or the *Fanout Completion Status* alert to the DM Server, depending upon the response handling selection made by the DM Server in the Fanout MO.

Step X: At some later time, the DM Server invokes the Get command against the Results sub-tree to the DM Gateway to get the aggregated results for the fanout command, within a context of a separate DM session.

Step Y: Optionally, the DM Server can execute the Fanout/<x>/Operation/Creategroup to populate a new group in the Gateway ConfigMO. The DM Gateway returns the URI of the new group to the DM Server.

8.3.2 Retention of Response Data

Once a fanout command has completed execution, the responsibility is on the DM Server to read the response data from the Results sub-tree. The DM Gateway will retain the response information for a certain period of time to provide the DM Server an opportunity to read the response. The DM Gateway MAY delete older responses, as per its local policy. It is, therefore, important for the DM Server to retrieve the fanout command response from the DM Gateway in a timely manner after the command has completed execution.

8.3.3 Fanout Forwarding and Response Processing

When the Exec operation is invoked on the Fanout/<x>/Operations/Start node, the DM Gateway sends the content of the Fanout/<x>/DMCommands node to each targeted End Device. To achieve this, the DM Gateway needs to properly construct the DM message to deliver the fanout commands to each End Device. The DM Gateway also needs to process the response received from the End Device prior to submitting results back to the DM Server.

8.3.3.1 Command Fanout

Before starting the fanout operation, the DM Server needs to properly configure the Fanout/<x>/DMCommands node according to the following rules:

- The value of the Fanout/<x>/DMCommands node MUST conform to the structure of the <SyncBody> element.
- The command identifier for each fanout command MUST be assigned by the DM Server.
- Each fanout command MUST use the proper node addressing scheme to guarantee the address validity across all targeted End Devices. For example, when the absolute target address of the fanout command is different across each End Device, the DM Server MUST use the virtual URI addressing.

After initiating the Fanout/<x>/Operations/Start operation, the DM Gateway MUST generate the DM message for each targeted End Device. The generated DM message MUST contain the fanout commands from the Fanout/<x>/DMCommands node.

The DM message generated by the DM Gateway MAY contain other DM Commands—strictly for the DM Gateway. In this case, the DM Gateway MUST NOT use the command identifiers already assigned by the DM Server for the fanout commands. Examples of DM Commands for the DM Gateway's own purposes are:

- Providing the <Status> element for the <SyncHdr> of the last message from the End Device.
- Issuing non-fanout commands to the End Device(s).
- Issuing alerts to the End Device(s).

The DM Gateway can deliver the generated DM message to the End Device either in the new DM Session or in the existing DM Session.

8.3.3.1.1 Illustrative Example

This section describes an illustrative example in which the DM Server requests the fanout of a Get command on the "./antivirus_data/version" node and the Replace command on the "./antivirus_data/reserve_exec" node. Note that in this example, it is assumed that all targeted End Devices have the same absolute target address.

The DM Server instantiates the Fanout MO and sets the value of the Fanout/<x>/DMCommands node as follows:

```
<SyncBody>
  <Get>
    <CmdID>1</CmdID>
    <Item>
      <Target>
        <LocURI>./antivirus_data/version</LocURI>
      </Target>
    </Item>
  </Get>
  <Replace>
    <CmdID>2</CmdID>
    <Item>
      <Target>
        <LocURI>./antivirus_data/reserve_exec</LocURI>
      </Target>
```

```
        <Data>2011-07-11T03:00Z</Data>
      </Item>
    </Replace>
</SyncBody>
```

The DM Gateway sends a DM Notification to establish a new DM Session for delivering the fanout commands. Alternatively, the DM Gateway can use an existing DM session, but this option is not described here. The Pkg#1 from the End Device to the DM Gateway is as follows:

```
<SyncML>
  <SyncHdr>...</SyncHdr>
<SyncBody>
  <Alert>
    <CmdID>1</CmdID>       <!-- Server-initiated session -->
    <Data>1200</Data>
  </Alert>
  <Replace>
    <CmdID>2</CmdID>      <!-- Replace for DevInfo -->
    ...
  </Replace>
  <Replace>
    <CmdID>3</CmdID>      <!-- Replace for DevDetail -->
    ...
  </Replace>
  <Alert>
    <CmdID>4</CmdID>       <!-- Client Event (could be Multiple) -->
    <Data>1224</Data>
    ...
  </Alert>
  <Alert>
    <CmdID>5</CmdID>       <!-- Generic Alert (could be Multiple) -->
    <Data>1226</Data>
    ...
  </Alert>
  <Alert>
    <CmdID>6</CmdID>       <!-- DM Tree Changed Alert -->
    <Data>1228</Data>
    ...
  </Alert>
</SyncBody></SyncML>
```

The DM Gateway generates the SyncML message for the End Device.

```
<SyncML>
  <SyncHdr>...</SyncHdr>
<SyncBody>
  <Status>
    <CmdID>3</CmdID><Cmd>SyncHdr</Cmd> <!-- Status for SyncHdr -->
  </Status>

  <Status>
    <CmdID>4</CmdID>
    <CmdRef>1</CmdRef>                <!-- Status for Server-inited
Session -->
  </Status>
  <Status>
    <CmdID>5</CmdID>
    <CmdRef>2</CmdRef>                                          <!--
Status for DevInfo -->
  </Status>
  ...
  <!-- More Status Here -->
  <Status>
    <CmdID>6</CmdID>
    <CmdRef>4</CmdRef>                            <!-- Status for Client
Event Alert -->
  </Status>
  <Get>
  <!-- fanout command Get -->
    <CmdID>1</CmdID>
    <Item>
      <Target>
        <LocURI>./antivirus_data/version</LocURI>
      </Target>
    </Item>
  </Get>
  <Replace>
  <!-- fanout command Replace -->
    <CmdID>2</CmdID>
    <Item>
      <Target>
        <LocURI>./antivirus_data/reserve_exec</LocURI>
      </Target>
      <Data>2011-07-11T03:00Z</Data>
    </Item>
  </Replace>
  ...                                                           <!-- More
Alerts Here -->
</SyncBody>
</SyncML>
```

8.3.3.2 Response Processing

On receiving the fanout response from the End Device, the DM Gateway MUST properly configure the Fanout/<x>/Results/<x> sub-tree to store the results. The node value of the DevResultXML and DevResultWBXML node MUST conform to the structure of the <SyncBody> element, as per the DM representation protocol [DMREPPRO]. The DevResultXML or DevResultWBXML node MUST store the results only for the fanout commands and MUST NOT store the <Status> element for the preceding <SyncHdr> and any results for commands issued locally by the DM Gateway.

The End Device can use the asynchronous response for some parts of the fanout commands. In this situation, the End Device will send the Generic Alerts to the DM Gateway after finishing the accepted commands. The DM Gateway MUST collect the

results from the Generic Alerts (that is, store the entire <Alert> node either in the DevResultXML or DevResultWBXML node). However, the DM Gateway MUST NOT store the <Status> elements for '(202) Accepted for processing', and also MUST NOT store any results if they are not related to the fanout commands.

In the case where the DM Server chooses to receive the *Result Aggregation Alert*, the DM Gateway MUST issue the alert after receiving responses from all of the targeted End Devices. In the case that the DM Gateway fails to receive responses from some of the End Devices, the *Result Aggregation Alert* MUST be sent after a certain timeout (left to implementation).

Additionally, the DM Gateway MAY send the *Result Aggregation Alerts* subsequently to report for delayed results from some of the End Devices. The subsequent *Result Aggregation Alerts* MAY also be sent to deliver periodic responses triggered by the fanout commands. On receiving the delayed or periodic responses, the DM Gateway MUST update either the DevResultXML or DevResultWBXML node accordingly. The subsequent *Result Aggregation Alerts* SHALL NOT include information that has already been reported in a previous *Result Aggregation Alert*, so as to minimize the redundant information from the DM Gateway to the DM Server.

8.3.3.2.1 Illustrative Example

This section describes an illustrative example in which the DM Gateway collects the results for the two fanout commands: Get "./antivirus_data/version" and Exec "./SCOMO/Download/PKG1/Operations/Download". The End Device responds synchronously for the Get and asynchronously for the Exec.

The Fanout/<x>/DMCommands node contains two DM Commands as follows, and the DM Gateway delivers them to the End Device(s).

```
<SyncBody>
  <Get>
    <CmdID>1</CmdID>
    <Item>
      <Target>
        <LocURI>./antivirus_data/version</LocURI>
      </Target>
    </Item>
  </Get>
  <Exec>
    <CmdID>2</CmdID>
    <Item>
      <Target>
        <LocURI>./SCOMO/Download/PKG1/Operations/Download</LocURI>
      </Target>
    </Item>
  </Exec>
</SyncBody>
```

The DM Gateway receives the responses for the fanout commands. The <Status> for the Exec command shows that it is an asynchronous response.

```
<SyncBody>
  <Status>                                                    <!--
Status for Get -->
    <CmdRef>1</CmdRef>
    <Cmd>Get</Cmd>
    <TargetRef>./antivirus_data/version</TargetRef>
    <Data>200</Data>
  </Status>
  <Results>                                                   <!--
Results for Get -->
    <CmdRef>1</CmdRef>
    <Item>
      <Source>
        <LocURI>./antivirus_data/version</LocURI>
```

```
        </Source>
        <Data>antivirus-inc/20010522b/5</Data>
      </Item>
    </Results>
    <Status>                                                    <!--
Async Response for Exec -->
      <CmdRef>2</CmdRef>
      <Cmd>Exec</Cmd>
      <Data>202</Data>
    </Status>
</SyncBody>
```

The DM Gateway receives the Generic Alerts for the result of the Exec command.

```
<Alert>
  <CmdID>2</CmdID>
  <Data>1226</Data>
    <Item>
      <Source>
        <LocURI>./SCOMO/Download/PKG1/Operations/Download</LocURI>
      </Source>
      <Target>
        <LocURI>./SCOMO/Inventory/Deployed/Component1</LocURI>
      </Target>
      <Meta>
        <Type>urn:oma:at:scomo:1.0:OperationComplete</Type>
        <Format>xml</Format>
      </Meta>
      <Data>
        <![CDATA[
          <ResultCode>1200</ResultCode>
          <Identifier>Component1ID</Identifier>
        ]]>
      </Data>
    </Item>
</Alert>
```

The previous status for Exec command is replaced with the Generic Alerts. The DM Gateway stores the results only for the fanout command in the DevResultXML node as follows:

```
<SyncBody>
    <Status>                                                    <!--
Status for Get -->
      <CmdRef>1</CmdRef>
      <Cmd>Get</Cmd>
      <TargetRef>./antivirus_data/version</TargetRef>
      <Data>200</Data>
    </Status>
    <Results>                                                   <!--
Results for Get -->
      <CmdRef>1</CmdRef>
      <Item>
        <Source>
          <LocURI>./antivirus_data/version</LocURI>
        </Source>
        <Data>antivirus-inc/20010522b/5</Data>
      </Item>
    </Results>
    <Alert>                                                     <!--
```

```
Generic Alert for Exec -->
     <CmdID>2</CmdID>
     <Data>1226</Data>
     <Item>
       <Source>
         <LocURI>./SCOMO/Download/PKG1/Operations/Download</LocURI>
       </Source>
       <Target>
         <LocURI>./SCOMO/Inventory/Deployed/Component1</LocURI>
       </Target>
       <Meta>
         <Type>urn:oma:at:scomo:1.0:OperationComplete</Type>
         <Format>xml</Format>
       </Meta>
       <Data>
         <![CDATA[
           <ResultCode>1200</ResultCode>
           <Identifier>Component1ID</Identifier>
         ]]>
       </Data>
     </Item>
   </Alert>
</SyncBody>
```

8.3.4 Proxy Secure Mechanism

This provides the secure mechanism for the DM Gateway to operate in proxy mode operation. This section includes sequences of server authorization using privilege MO and provides end –to -end security in adding /modifying, when at least one DM Server account available in End devices.

8.3.4.1 DM Privilege Secure Sequence

This section provides the success and failure sequences of the DM Server Authorization for performing Management commands.

Success Sequence:- Gateway Checks Privilege ACL for Server Authorization as shown in Figure 14.

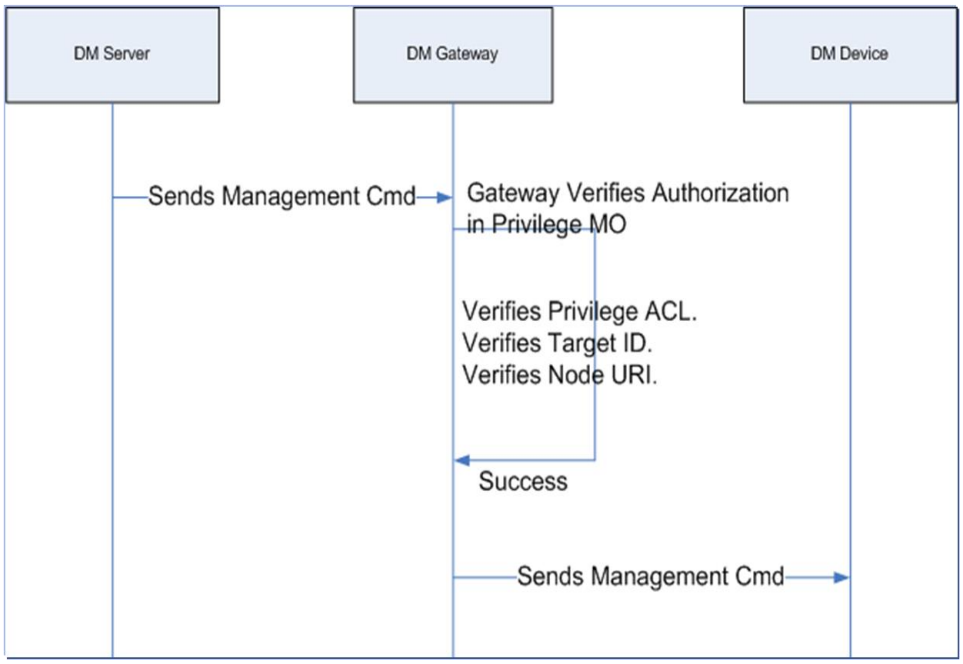

Figure 14: DM Privilege Secure Sequence -- Success

Failure Sequence:- Gateway Checks Privilege ACL for Server Authorization as shown in Figure 15.

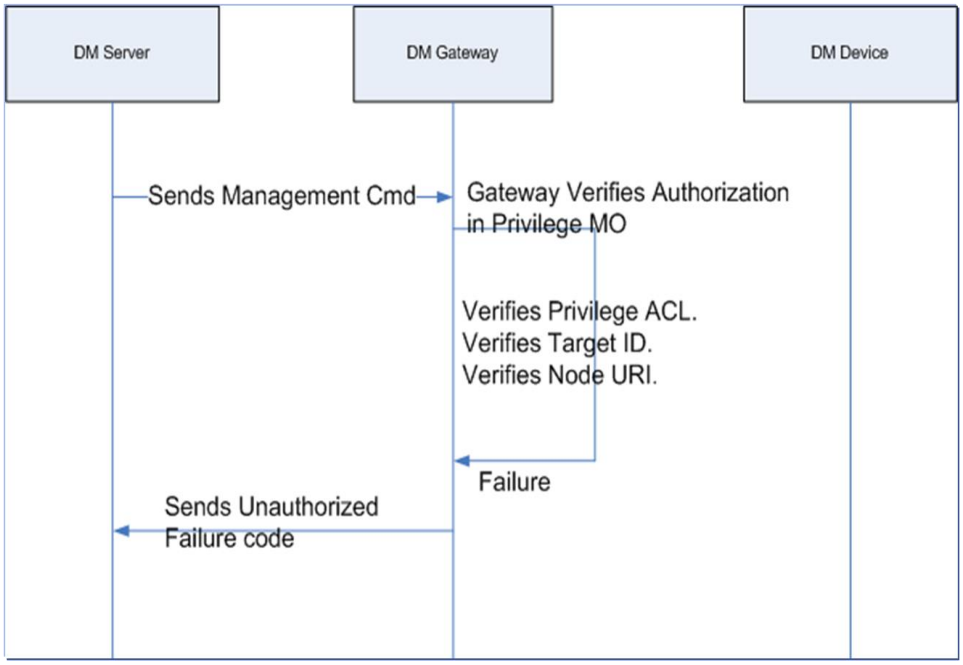

Figure 15: DM Privilege Secure Sequence -- Failure

8.3.4.2 Add or Modify Privilege ACL in Secure way

Figure 16 and other below figures shows the sequence for adding or modifying privilege ACL in secure way, when at least one Primary DM server account present on the device or group. This secure mechanism is used when multiple DM Servers that want to manage a device or group under the DM Gateway with at least one primary DM server account on the End

Device. In this scenario, provides End-to-End security by receiving a device management command to Privileges nodes in Gateway Config MO from primary server to add/modify privilege ACL – in order to add permissions for the secondary servers, which does not have access right on certain resources of end devices. In this case the DM Gateway proceeds with the following steps.

Success Case for Device:

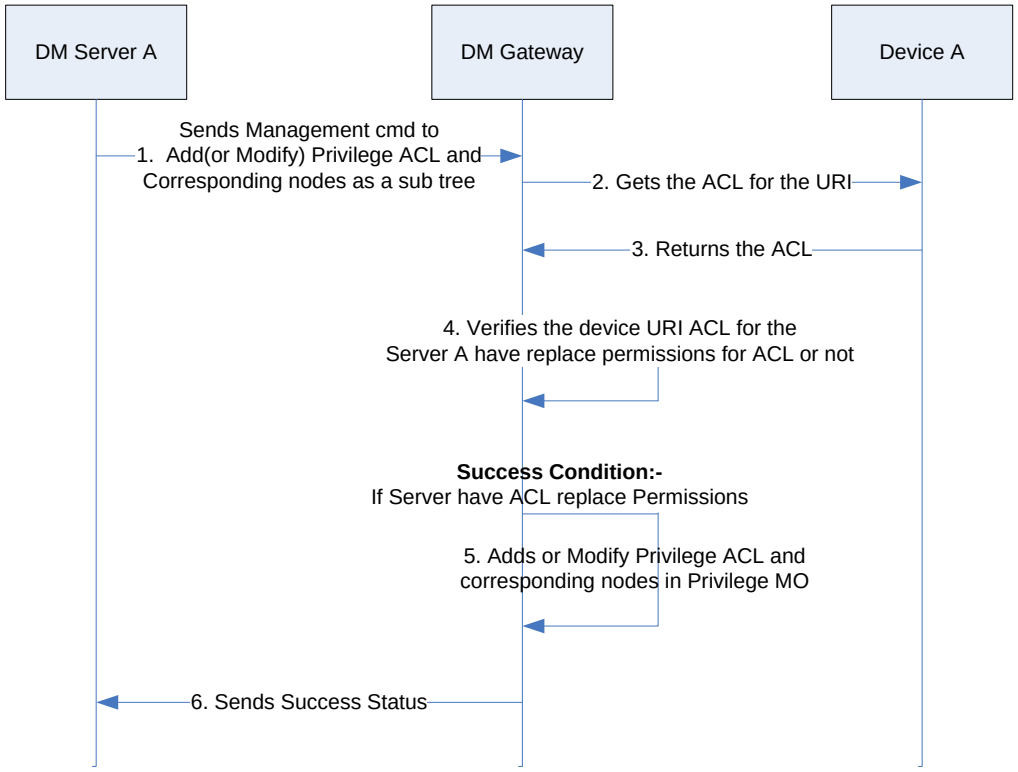

Figure 16: Addition or Modification of Privilege ACL in Secure Way for Device - Success.

Step 1: The DM server A sends management command to add or modify the Privilege ACL and the corresponding nodes on the DM Gateway.

Step 2: The DM Gateway gets the ACL for the corresponding URI of the End Device.

Step 3: The End Device A returns the ACL queried by the DM Server.

Step 4: The DM Gateway verifies the Device A URI ACL and checks whether Server A have replace permissions for the Device URI ACL or not.

Step 5: If the verification proved success that the Server A has replace permissions on ACL of the Device URI, Gateway adds or modifies Privilege ACL successfully

Step 6: Gateway sends the success status to the server A

Figure 17 shows the Failure sequence for adding or modifying privilege for End Device.

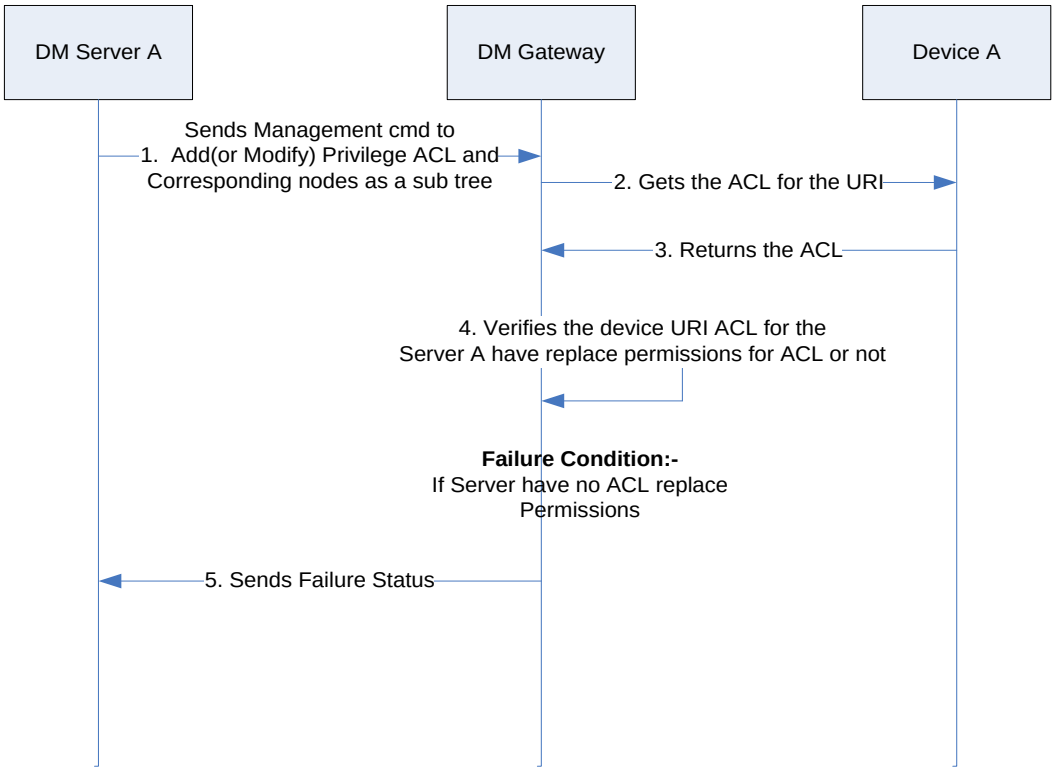

Figure 17: Addition or Modification of Privilege ACL in Secure Way for Device - Failure.

Step 1: The DM server A sends management command to add or modify the Privilege ACL and the corresponding nodes on the DM Gateway.

Step 2: The DM Gateway gets the ACL for the corresponding URI of the End Device.

Step 3: The End Device A returns the ACL queried by the DM Server.

Step 4: The DM Gateway verifies the Device A URI ACL and checks whether Server A have replace permissions for the Device URI ACL or not.

Step 5: If the verification proved failure, Gateway sends the failure status to the server A

Figure 18 shows the Success sequence for adding or modifying privilege for the Group.

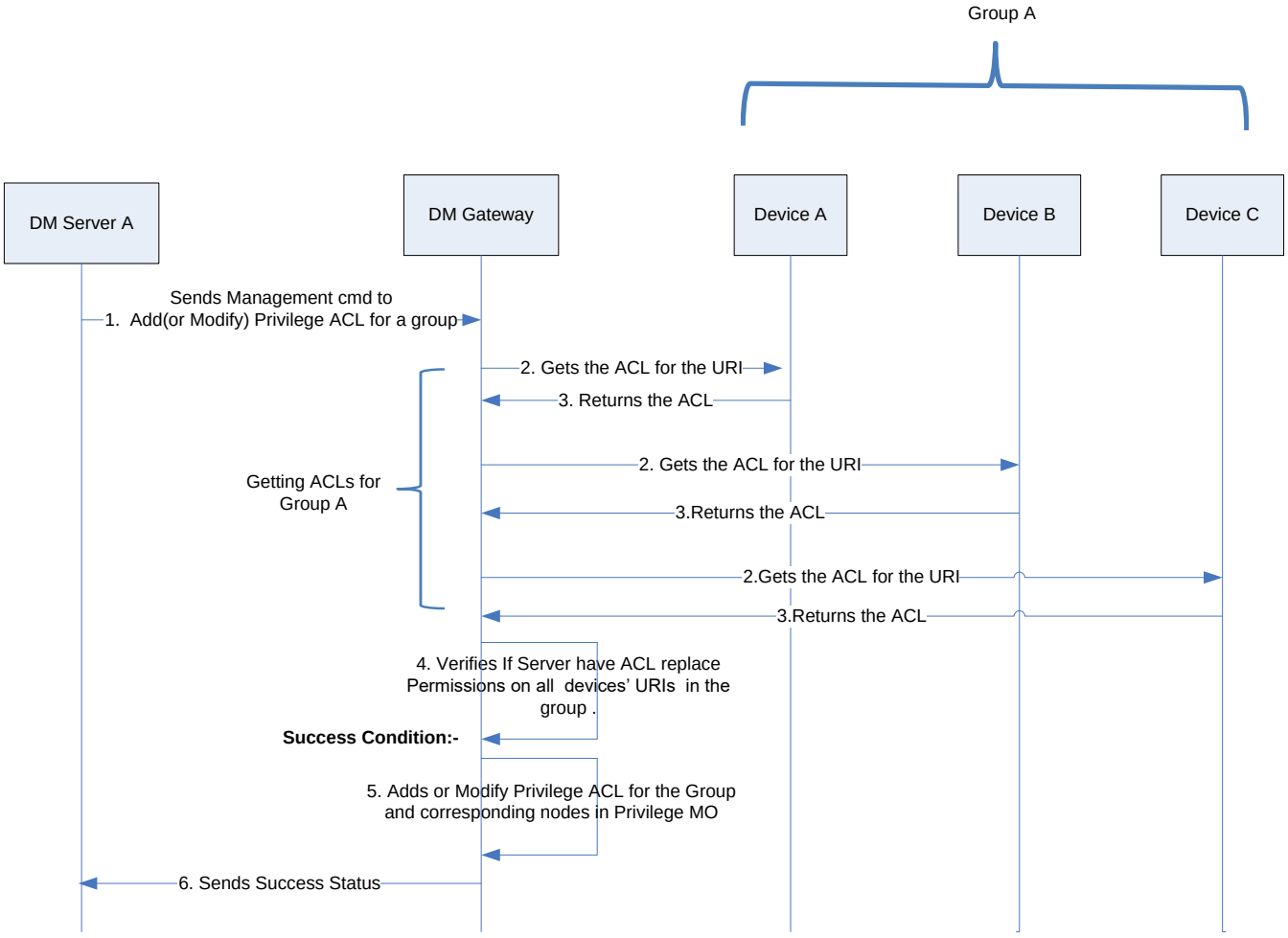

Figure 18: Addition or Modification of Privilege ACL in Secure Way for Group - Success

Step 1: The DM server A sends management command to add or modify the Privilege ACL and the corresponding nodes on the DM Gateway for Group.

Step 2: The DM Gateway gets the ACLs for the corresponding URIs of all the End Devices in the Group (Group ID).

Step 3: The End Devices A, B and C of the Group returns the ACLs.

Step 4: The DM Gateway verifies all the Devices URIs' ACLs of the Group and checks whether Server A have replace permissions or not.

Step 5: If the verification proved success that the Server A have replace permissions on ACLs of all corresponding Devices' URIs in associated Group, Gateway adds or modifies Privilege ACL successfully for the Group.

Step 6: Gateway sends the success status to the server A

Figure 19 shows the Failure sequence for adding or modifying privilege for the Group.

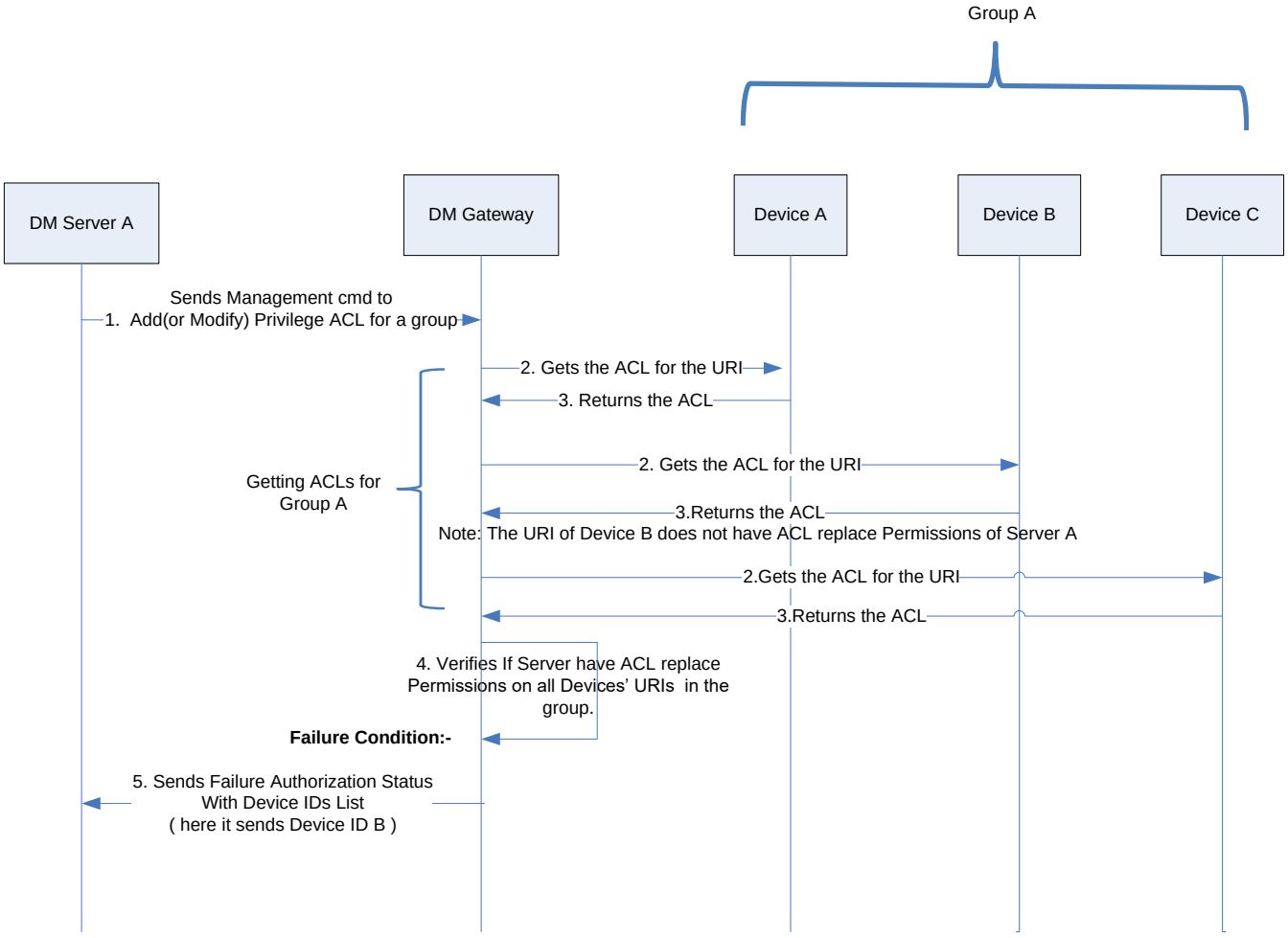

Figure 19: Addition or Modification of Privilege ACL in Secure Way for Group - Failure

Step 1: The DM server A sends management command to add or modify the Privilege ACL and the corresponding nodes on the DM Gateway for Group.

Step 2: The DM Gateway gets the ACLs for the corresponding URIs of all the End Devices in the Group (Group ID).

Step 3: The End Devices A, B and C of the Group returns the ACLs.

Step 4: The DM Gateway verifies all the Devices URIs' ACLs of the Group and checks whether Server A have replace permissions or not. (Here example Device B URI does not have replace permissions)

Step 5: If the verification proved Failure, Gateway sends the failure status with failed device IDs list to the server A. (here in this case it sends Device B ID)

8.4 Realizing Adaptation Mode Functionality (Informative)

This informative section describes a few approaches for realizing the Protocol Adaptation Mode. Other approaches for realizing the Adaptation Mode are not precluded.

8.4.1 Adaptation Using Fanout MO

This approach relies on the Fanout MO with the added capability of being able to adapt OMA DM messages to the native management protocol supported by the End Devices.

The following steps constitute a typical flow for the Adaptation Mode using this approach, as shown in Figure 20:

1. The DM Server instantiates the Fanout MO on the DM Gateway within the context of a DM session between the DM Server and the DM Gateway.
2. The DM Server sets the DM command in the *DMCommands* node of this MO instance. The DM Server then issues an Exec command on the *Operations/Start* node of this MO instance.
3. The DM Gateway adapts the DM command to the native management protocol of the End Device and forwards the adapted command to the End Device
4. Upon receiving the response from the End Device, the DM Gateway adapts the response to the DM protocol and sets it in the *DevResultXML* or *DevResultWBXML* node, for retrieval by the DM Server.

One advantage with this approach is that the DM Server is completely oblivious to the management protocol running between the DM Gateway and the End Device. Thus, the OMA DM End Devices in Proxy Mode and non-OMA DM End Devices in Adaptation Mode are treated exactly the same from DM Server's perspective.

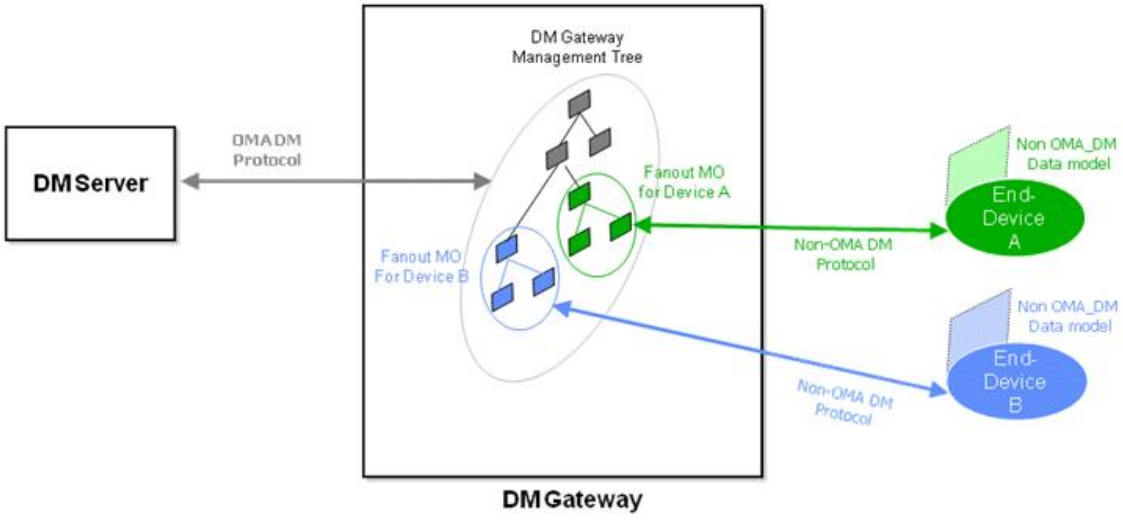

Figure 20: GwMO Adaptation Mode realisation with Fanout MO

8.4.2 Adaptation Using Protocol Encapsulation

As the DM Gateway cannot always support adaptation of all possible native management protocols of End Devices, an alternative approach is described here.

Using this approach, the Management Authority can use the OMA DM infrastructure to easily manage End Devices with native management protocols that are not supported by the DM Gateway in deployment. Here, the protocol adaptation functionality is provided by the management authority using a non-OMA DM server, operating in conjunction with an OMA DM Server.

The following steps constitute a typical flow for the Adaptation Mode using this approach:

1. The Management Authority creates the non-OMA DM command message and makes it available to the OMA DM Server, via some unspecified mechanism.

2. The OMA DM Server instantiates the Fanout MO on the DM Gateway within the context of a DM session between the OMA DM Server and the DM Gateway. This MO instance contains the *NonDMCommands* and *DevResultBIN* nodes.
3. The DM Server sets the non-OMA DM command as the value of the *NonDMCommands* node of this MO instance.
4. The DM Server issues an Exec command on the Operations/Start node of this MO instance.
5. The DM Gateway forwards the non-OMA DM command to the End Device. It is assumed that the DM Gateway can communicate with the End Devices at least in the transport layer (for example, Ethernet), even if the non-OMA DM protocol is not supported between the DM Gateway and non-OMA DM End Devices.
6. Upon receiving the response from the End Device, the DM Gateway sets it in the *DevResultBin* node for retrieval by the Management Authority.

One advantage with this approach is that the protocol adaptation operation is the responsibility of the Management Authority, which may include a separate non-OMA DM Server communicating with the OMA DM Server. The interface between the non-OMA DM Server and the OMA DM Server is out-of-scope of this specification.

As can be clearly seen, the non-OMA DM management commands are delivered to the DM Gateway using the OMA-DM infrastructure. The OMA-DM protocol has no other role in this approach.

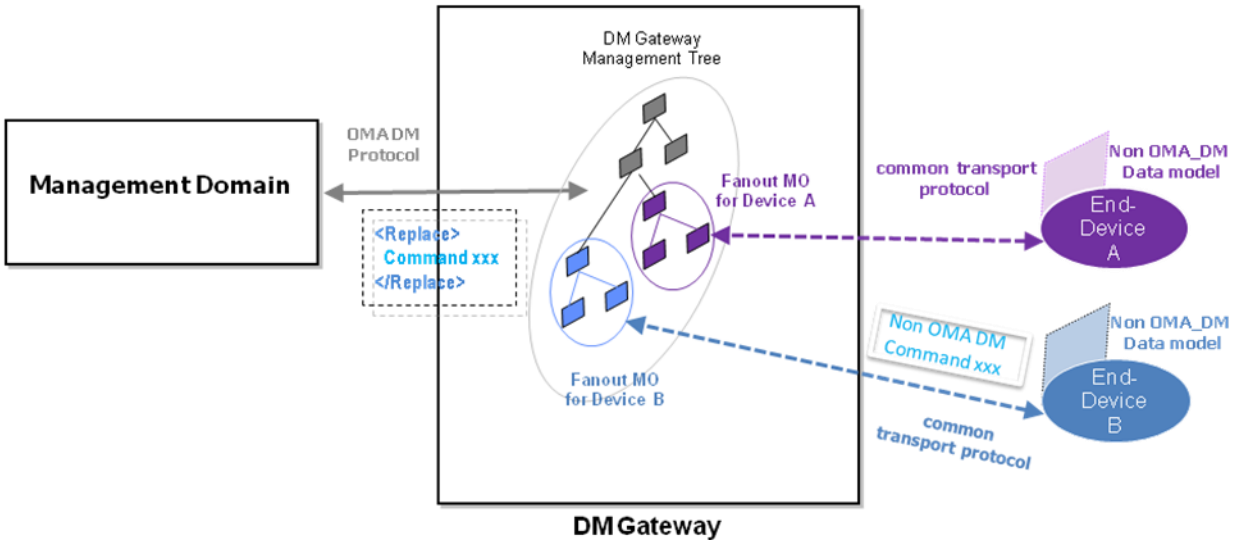

Figure 21: GwMO Adaptation Mode Realisation Using Protocol Encapsulation

8.4.3 Adaptation with DM Gateway in Origin Server Role

In some instances, the DM Gateway can play the role of the *Origin Server* by hosting the Management Tree of the End Device. In other words, the DM Gateway presents each non-OMA DM End Device as a logical OMA DM Device to the DM Server, as shown in Figure 22. As can be seen, the DM Server directly invokes commands on the Management Tree of the End Device. The DM Gateway adapts the DM commands to the native management protocol of the End Device and forwards the adapted commands to the End Device. Upon receiving the response from the End Device, the DM Gateway adapts the response to DM format, updates the DM Tree for the End Device (if needed), and forwards the adapted response to the DM Server.

In this approach, the DM Gateway has to deal with the extra overhead of maintaining the Management Tree for each End Device, which may become an issue if the number of attached non-OMA DM End Devices is excessive.

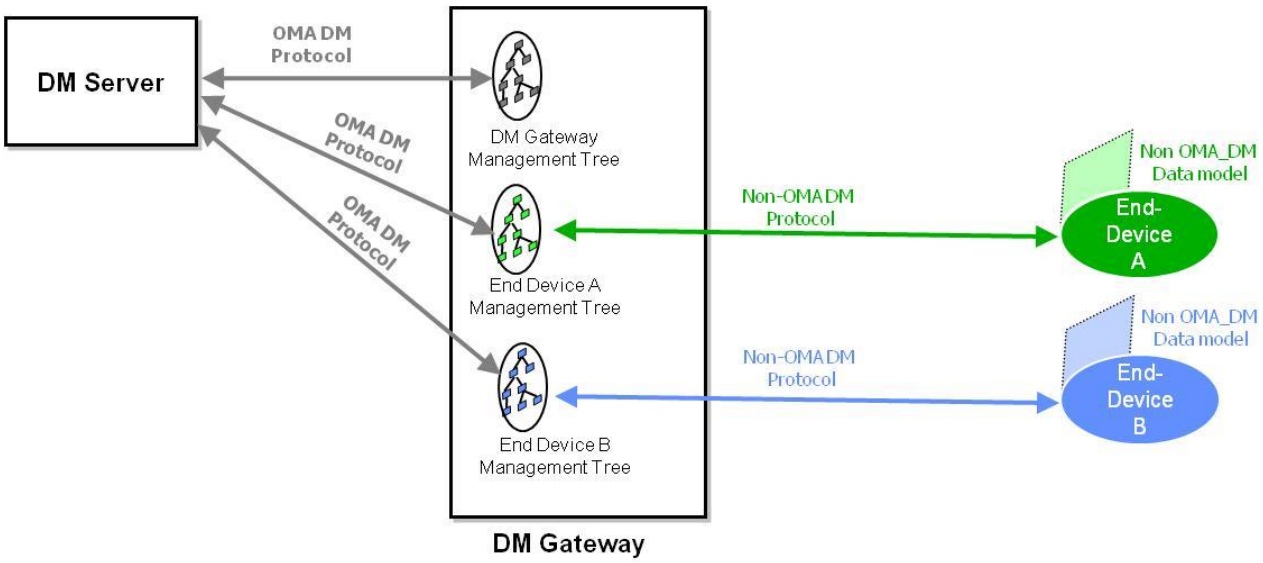

Figure 22: GwMO Adaptation Mode Realization with Gateway as Origin Server

8.4.3.1 Protocol Adapter GwMO-5 Interface

According to [GwMO_AD_v1.1], the DM Gateway SHALL include a Protocol Adapter component, and the Protocol Adapter SHALL support GwMO-5 interface.

The GwMO-5 interface allows the DM Gateway to interact with Protocol Adapter component in order to send commands and exchange data structures needed for performing the OMA DM protocol to non-OMA DM protocol adaptation.

In order to specify an implementation-independent interface, an IDL ([OMGIDL]) definition of GwMO-5 is given.

```
module OMADMGateway {

	String moduleName;
	enum actionType { GET, REPLACE, GET_TREE, ADD, DELETE };

	// Optional: NULL if actionType = GET_TREE; else depends from
	// moduleName and actionType invoked.
	String actionName;

	// Optional: the value for paramName depends from actionType and actionName invoked.
	String paramName;
	typedef sequence<String> ListParameter;
	typedef sequence<String> ListTarget;

	// Optional: NULL if actionType = GET_TREE
	String vendorProt;

	interface OMADMGatewayAPI {
		void sendCommand(	in String moduleName, in String actionType, in String
actionName, in String paramName,

							in ListParameter parameters,
							// This list can be empty, single-value or
comma-separated multi-values
```

```
                                                in ListTarget targets,
                                                //  Optional,  target  Devices  list  (comma-
separated)
                                                // empty if <action_type>=”GET_TREE”)

                                                in String vendorProt,
                                                in String vendorVersion);
        };

};
```

For example, for a DM Gateway to Building Automation End Devices, it can be:

moduleName = LIGHT (or CAMERA, SHUTTER, HVAC…)

actionName = LIGHT_MODE

paramName = BLINK (or DIMMER…)

vendorProt = KNX (or OpenWebNet…)

8.5 Bootstrapping

8.5.1 Bootstrapping the End Device to the DM Gateway

The DM Bootstrap specification [DMBOOT] lists four possible approaches to the DM Bootstrap process: customized bootstrap, server initiated bootstrap, bootstrap from SmartCard, and client initiated bootstrap. In theory these bootstrapping approaches can work for DM Gateway Bootstrapping on End Devices as well. However, in practice some of these bootstrapping approaches may not be feasible for End Devices. For example, some of the End Devices may operate in private networks, which are outside the control of traditional service providers. For this reason, there may not be enough cost justification for customized bootstrap and SmartCard bootstrap. Server initiated bootstrap on a blank End Device is also a big challenge because the Device may not have a globally routable address that can be used by some push mechanisms (for example, OMA Push) to bootstrap the End Device.

This specification describes two approaches for Gateway Bootstrapping on the End Device. These two approaches are described in the following subsections. The DM Gateway MUST support at least one of these approaches. It needs to be noted that other approaches for Gateway Bootstrapping on the End Device are not precluded.

There are some scenarios in which the End Device has to distinguish between a DM Server and a DM Gateway. To handle such scenarios, the value of the ‘Ext/oma_gwmo/ServerType’ node in DMAcc MO indicates whether the entity the Device interacting with is a DM Server or a DM Gateway.

8.5.1.1 DM Server Assisted Bootstrapping

In this approach, a previously bootstrapped DM Server bootstraps the End Device to the DM Gateway, as shown in Figure 23 and Figure 24. The DM Server bootstraps the End Device to the DM Gateway the very first time a DM session is established between the End Device and the DM Server, or whenever the End Device is detected in a new location. The installation of DM Gateway credentials on the End Device is performed using the normal bootstrapping process [DMBOOT]. The installation of End Device credentials on the DM Gateway is performed through the Gateway Config MO. As is clear from Figure 23 and Figure 24, the association between the End Device and the DM Gateway is established by the DM Server.

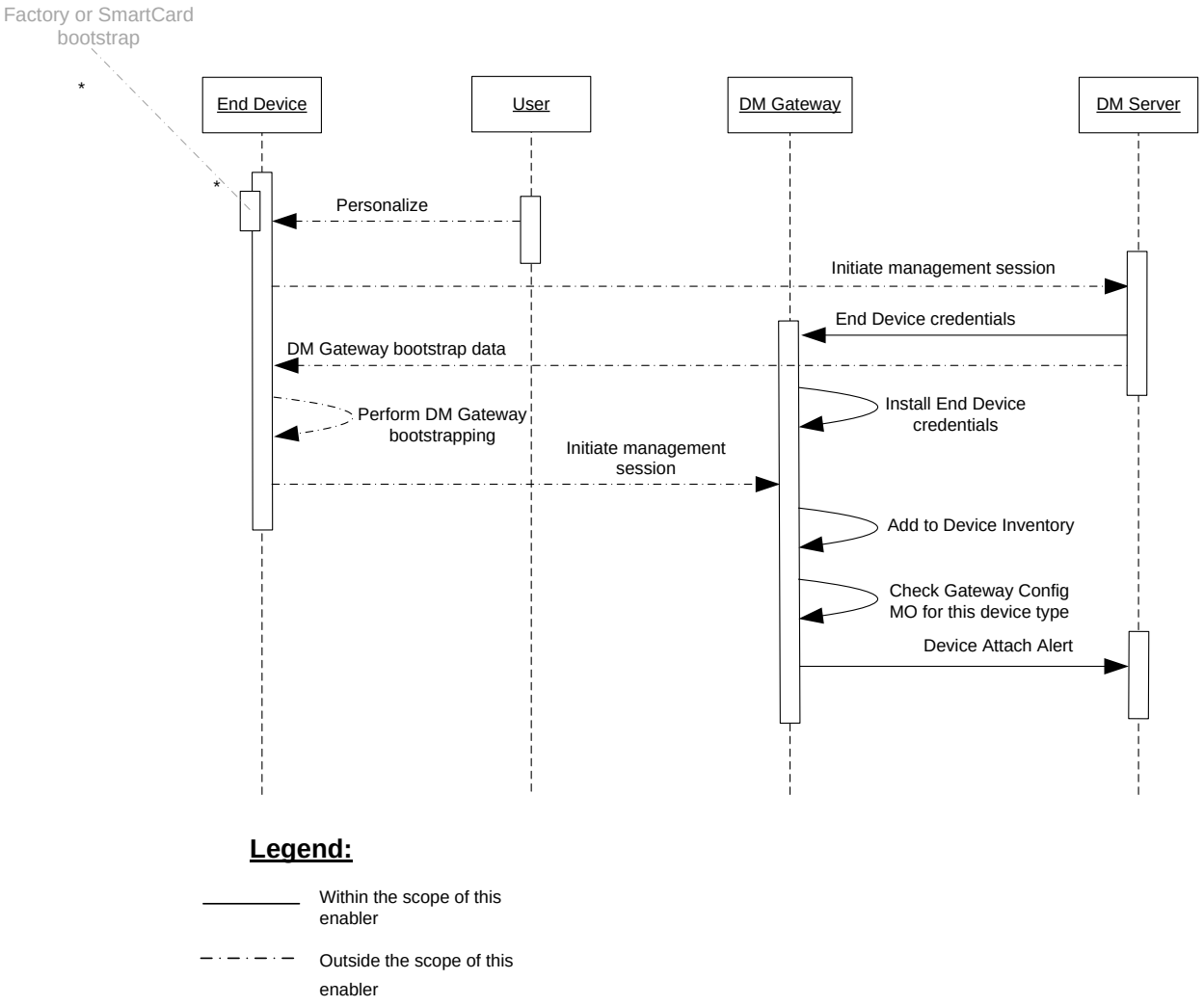

Figure 23: DM Server Assisted Gateway Bootstrapping

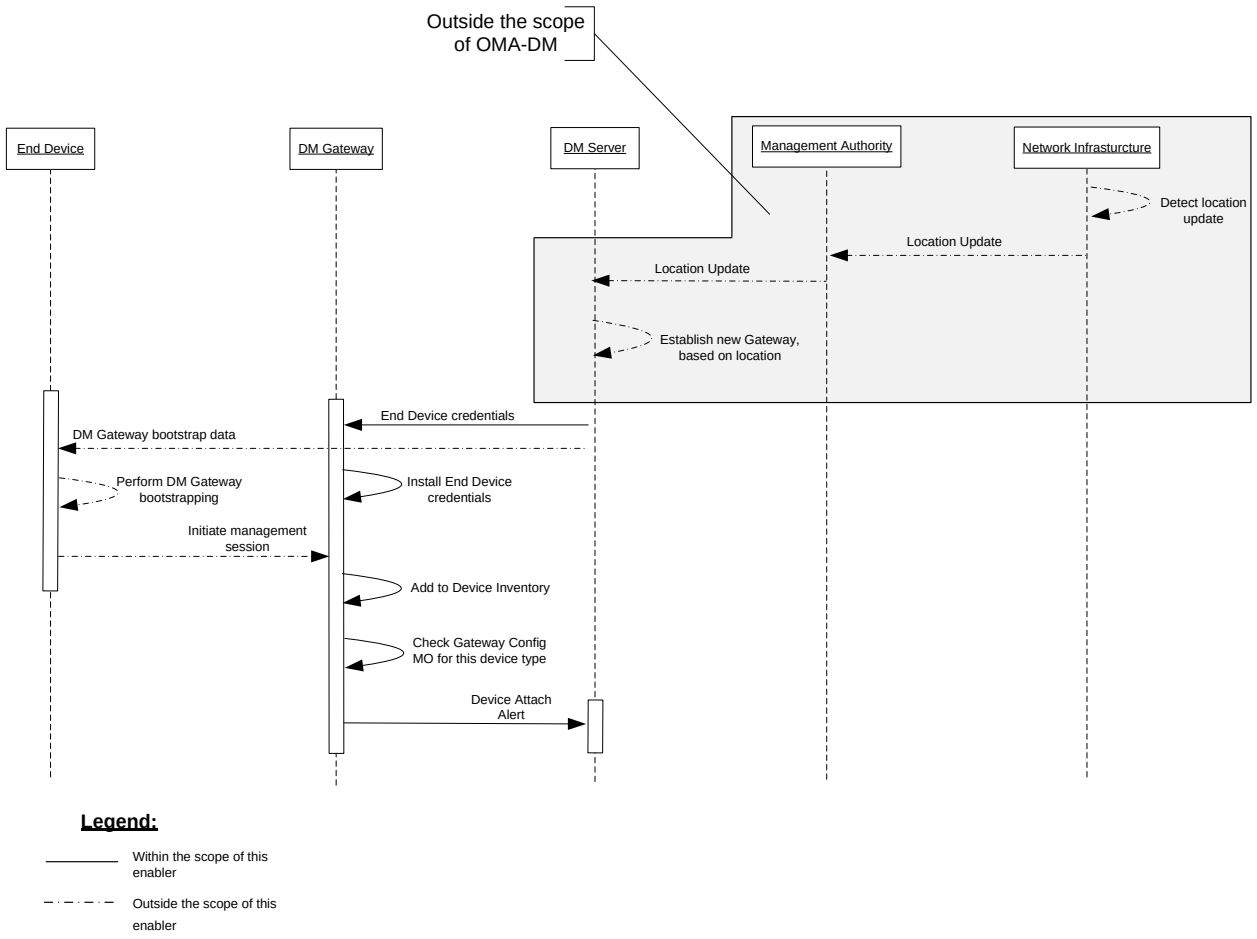

Figure 24: DM Gateway Bootstrapping Following Device Location Update

8.5.1.2 Local DM Bootstrap Server Assisted Bootstrapping

In this approach, the End Device obtains the DM Gateway Bootstrap from a local DM Bootstrap Server, as described in [DMBOOT].

It is clear from Figure 25 that the association between the End Device and the DM Gateway gets established dynamically. The DM Server learns about this association upon receiving the Device Attach Alert from the DM Gateway.

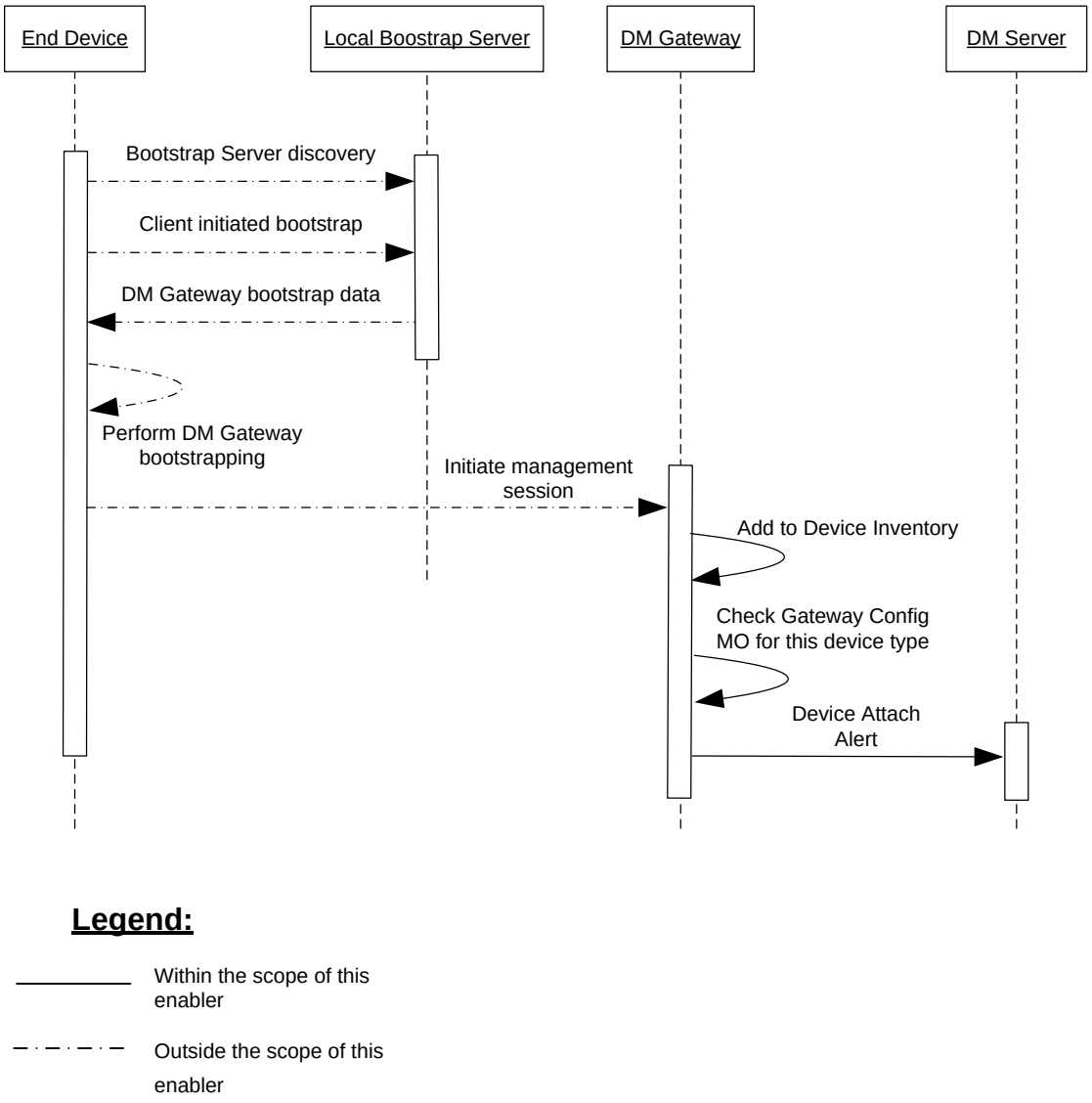

Figure 25: Local DM Bootstrap Server Assisted Gateway Bootstrapping

8.5.2 Bootstrapping the DM Gateway

The DM Gateway SHALL be bootstrapped in accordance with [DMBOOT].

8.5.3 Pre-bootstrapped End Devices

This specification discusses various scenarios for End Device bootstrapping.

- The End Device is only bootstrapped to the DM Gateway.
- The End Device is only bootstrapped to a subset of DM Servers to which a DM Gateway is bootstrapped.
- The End Device is pre-bootstrapped to some DM Servers to which the DM Gateway is not bootstrapped. For example, an End Device may be factory bootstrapped to the vendor's DiagMon DM Server, while the DM Gateway to which the End Device is attached is completely unaware of this DM Server. Another example is a case where a nomadic device

moves to a different location, where the DM Gateway is not bootstrapped to a DM Server to which the End Device had been previously bootstrapped.

This enabler supports the following three possibilities for the DM Gateway to handle End Devices that are pre-bootstrapped to other DM Servers.

- The DM Gateway ignores all pre-bootstrapped information on the End Device
- The DM Gateway gets bootstrapped to any DM Server to which the End Device is bootstrapped but the DM Gateway is not
- The DM Gateway does not get bootstrapped to any DM Server to which the End Device is bootstrapped but the DM Gateway is not. However, a DM Notification messages, with the Push Extension Header, is forwarded to the End Device if it originates from such a DM Server

These possibilities are left to the DM Gateway implementation and they are discussed in the following subsections.

8.5.3.1 Ignoring Pre-bootstrapped Information

In this case the DM Gateway only allows the End Device to be managed by a DM Server to which it is bootstrapped itself. If the End Device has a publicly routable address that is known to the DM Server, the DM Server can directly communicate with the End Device, bypassing the DM Gateway completely.

8.5.3.2 Bootstrapping to End Device's DM Server

In this case the DM Gateway proceeds with the following steps to trigger the bootstrapping of the DM Gateway to the End Device's DM Server.

Step 1: The End Device retrieves the bootstrap package from the local Bootstrap Server.

Step 2: The End Device installs the bootstrap package.

Step 3: The End Device reads the value of the 'Ext/oma_gwmo/ServerType' node in DMAcc MO to recognize the DM Server as a DM Gateway.

Step 4: The End Device sends a *Bootstrapped DMS List* Generic Alert (section7.3) to the DM Gateway. This Generic Alert lists all the DM Servers to which the End Device has been previously bootstrapped.

Step 5: The DM Gateway checks the list of DM Server ID(s) and identifies the DM Servers to which the DM Gateway is not already bootstrapped.

Step 6: The DM Gateway starts the client initiated bootstrap, as defined in section 5.1.2.4 of [DMBOOT].

Figure 26 shows the work flow.

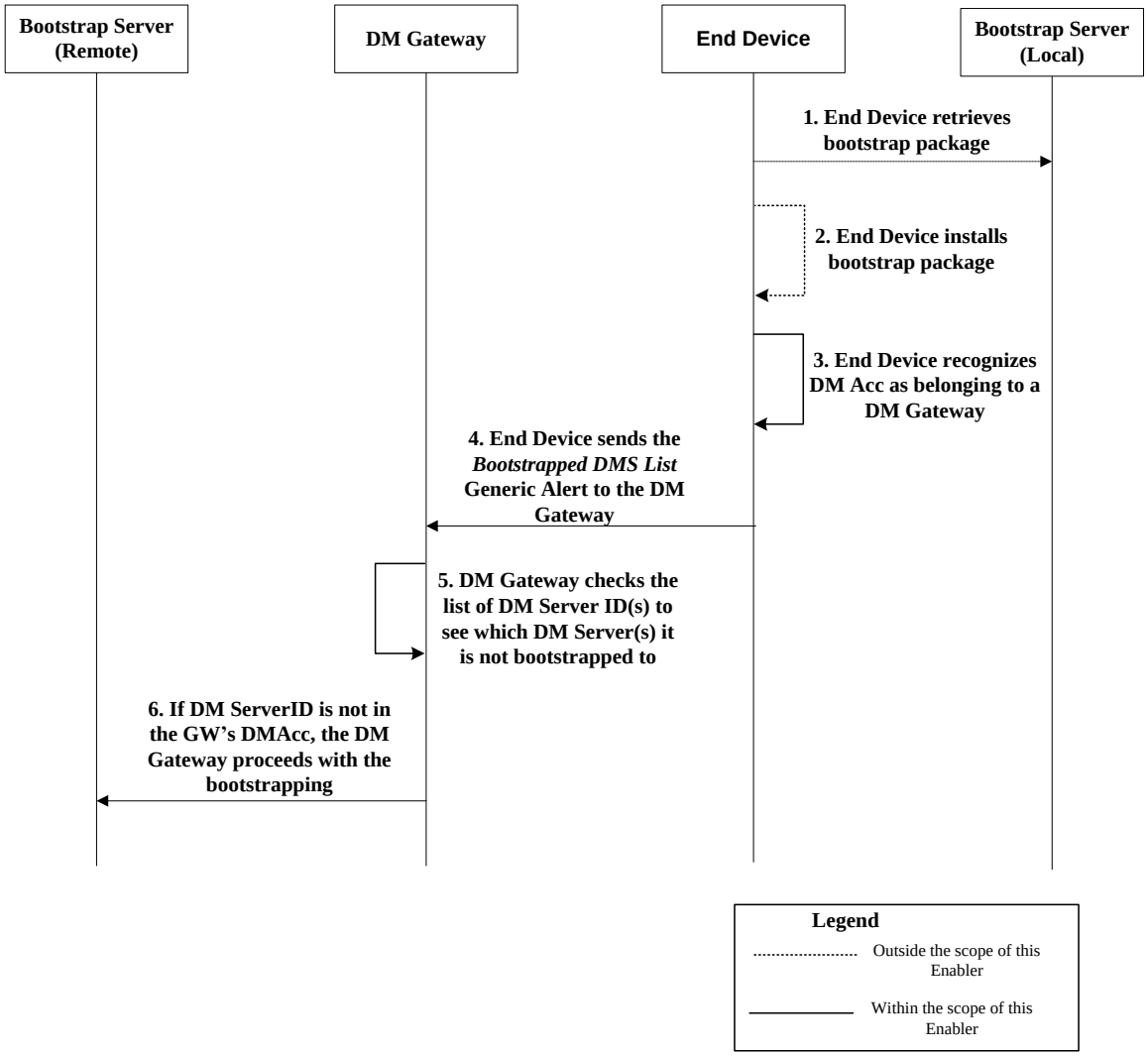

Figure 26: DM Gateway Bootstrapping to End Device's DM Server

8.5.3.3 Allowing Transparent Mode Operation for End Device's DM Server

In this case the DM Gateway proceeds with the following steps to allow the Transparent Mode operation for the End Device's DM Server, without getting bootstrapped to the DM Server.

Step 1: This step is the same as step 1 in section 8.5.3.2.

Step 2: This step is the same as step 2 in section 8.5.3.2.

Step 3: This step is the same as step 3 in section 8.5.3.2.

Step 4: This step is the same as step 4 in section 8.5.3.2.

Step 5: The End Device sends the *Associated Gateway* Generic Alert to that DM Server. This alert provides the publicly routable address of the DM Gateway to the DM Server.

Step 6: This step is the same as step 5 in section 8.5.3.2.

Step 7: The DM Gateway establishes an internal mapping between the DM Server ID and the End Device ID. This enables the DM Gateway to provide the Transparent Mode operation for the End Device for a DM Notification message, with Push Extension Header, which originates from the DM Server, even though the DM Gateway is not bootstrapped to the DM Server.

Figure 27 shows the work flow.

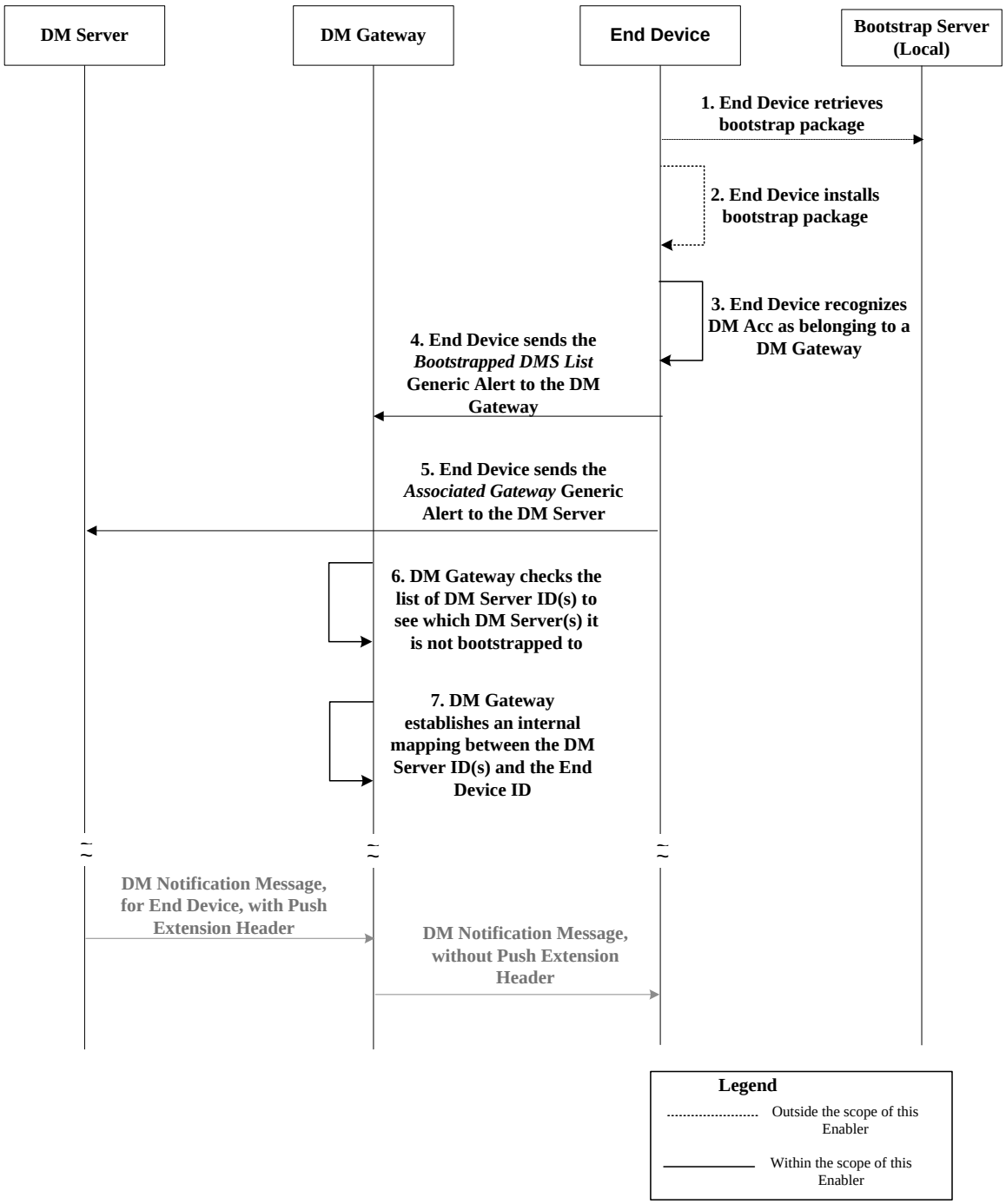

Figure 27: DM Gateway supporting the Transparent Mode for the End Device's DM Server

8.6 Image Distribution

To efficiently distribute images (for example, Delivery Package for SCOMO) to End Devices, the DM Server utilizes the Image Inventory MO resident on the DM Gateway. In this section, image distribution examples using the Image Inventory MO are shown. Although these illustrative examples show the package delivery of SCOMO, the Image Inventory MO can work with any other enablers that deal with image distributions.

The DM Server can either deliver the image to the DM Gateway by using an alternative download mechanism or by using the OMA-DM protocol. In the latter case, the DM Server performs a Replace operation on the <x>/Images/<x>/Data node, which contains the binary data for the image data.

8.6.1 Image Distribution in Proxy Mode

Figure 28 shows the flow for the image distribution with the DM Gateway operating in the Proxy Mode. The various steps in the flow are listed below.

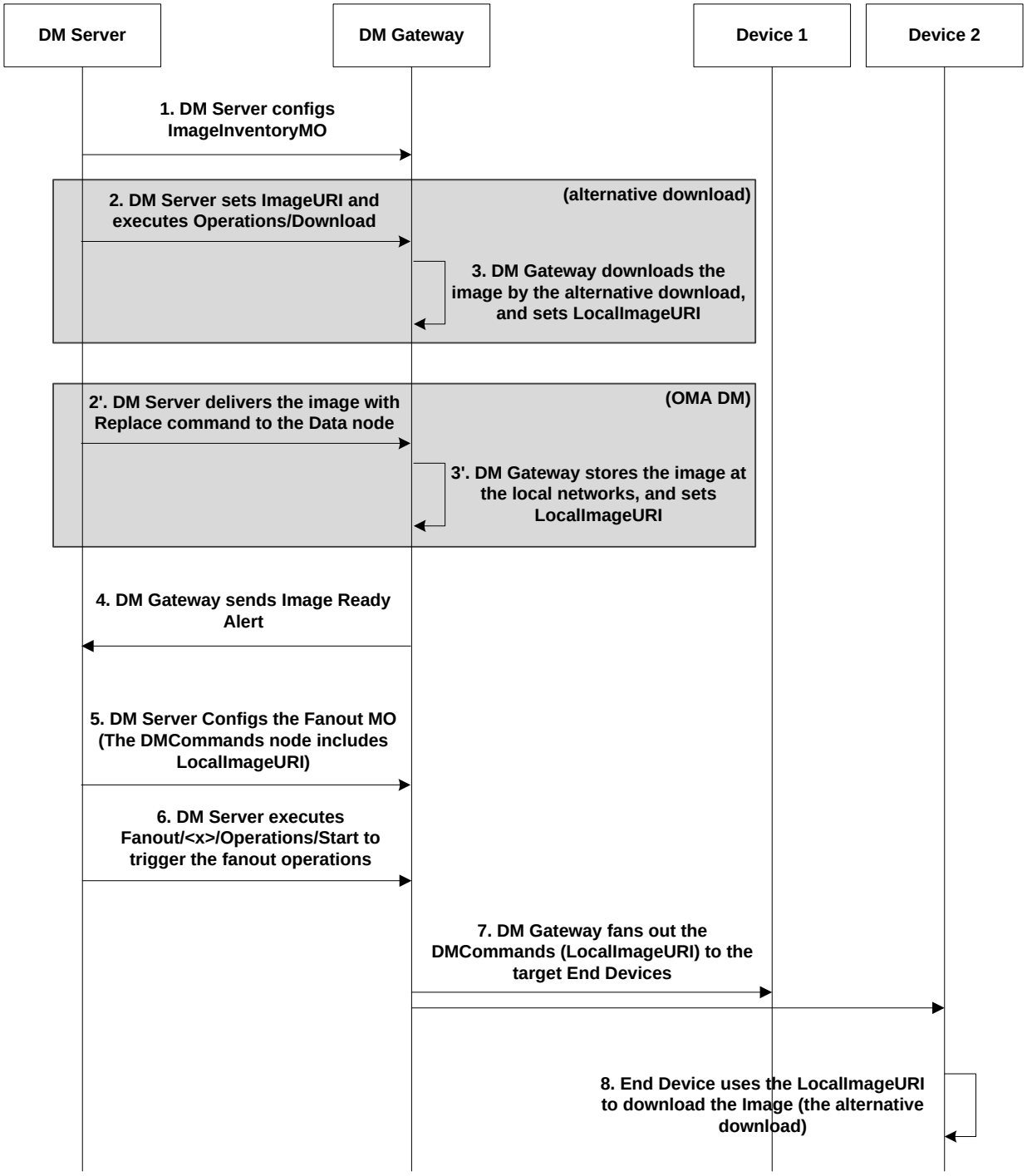

Figure 28: Image Distribution in Proxy Mode

Step 1: The DM Server configures the Image Inventory MO to distribute an image (e.g. Delivery Package for SCOMO) to multiple End Devices.

Step 2: (Alternative download case) The DM Server sets the value for the ImageURI node. The DM Server subsequently invokes the Exec operation on the <x>/Images/<x>/Operations/Download node to trigger the DM Gateway to download the image.

Step 3: (Alternative download case) The DM Gateway downloads the image. The image is stored on the DM Gateway itself, or on a storage server on the local network. After completely downloading the image, the DM Gateway sets the LocalImageURI node (e.g. http://fe80::202:b3ff:fe1e:8329/scomo/package123).

Step 2': (OMA DM case) For delivering the image, the DM Server uses the Replace command to the <x>/Images/<x>/Data node.

Step 3': (OMA DM case) On completely receiving the image, the DM Gateway sets the LocalImageURI node (e.g. http://fe80::202:b3ff:fe1e:8329/scomo/package123). For this, the DM Gateway can store the image on the DM Gateway itself or at a local storage server.

Step 4: The DM Gateway sends the Image Ready Alert. The Image Ready Alert contains the value of the LocalImageURI node.

Step 5: The DM Server instantiates the Fanout MO or updates an existing Fanout MO instance on the DM Gateway's Management Tree. When the DM Server sets the value for the Fanout/<x>/DMCommands node, it refers to the local image URI, provided by the DM Gateway previously in the Image Ready Alert, for initiating download on the End Device.

The example below shows the inclusion of the local image URI in the Fanout/<x>/DMCommand node prepared at the DM Server for forwarding to the End Device, via the DM Gateway:

```
<Add>
  <CmdID>1</CmdID>
  <Item>
    <Target><LocURI>./GWMO/Fanout/fo1/DMCommands</LocURI></Target>
    <Data>
      <![CDATA[          <!-DMCommands is inlined within <![CDATA[..]]-->
       <Add>
         <CmdID>1</CmdID>
         <Item>
           <Target><LocURI>./SCOMO/Download/Pkg1</LocURI></Target>
         </Item>
       </Add>
       <Add>
         <CmdID>2</CmdID>
         <Item>
           <Target><LocURI>./SCOMO/Download/Pkg1/PkgID</LocURI></Target>
           <Data>Package123</Data>
         </Item>
       </Add>
       <Add>
         <CmdID>3</CmdID>
         <Item>
           <Target><LocURI>./SCOMO/Download/Pkg1/PkgURL</LocURI></Target>
           <Data>http://fe80::202:b3ff:fe1e:8329/scomo/package123</Data> <!-
```

```
  Local URI-->
          </Item>
        </Add>
      ]]>
    </Data>
   </Item>
</Add>
```

Step 6: The DM Server executes the Fanout/<x>/Operations/Start node to trigger the fanout procedure.

Step 7: The DM Gateway fans out the Fanout/<x>/DMCommands to the target End Devices.

Step 8: The End Device uses the local image URI (e.g. http://fe80::202:b3ff:fe1e:8329/scomo/package123) to download the image by the alternative download.

8.6.2 Image Distribution in Transparent Mode

Figure 29 shows the flow for the image distribution with the DM Gateway operating in the Transparent Mode. The various steps in the flow are listed below. Note that below flow considers two End Devices, but it can cover multiple End Devices as well.

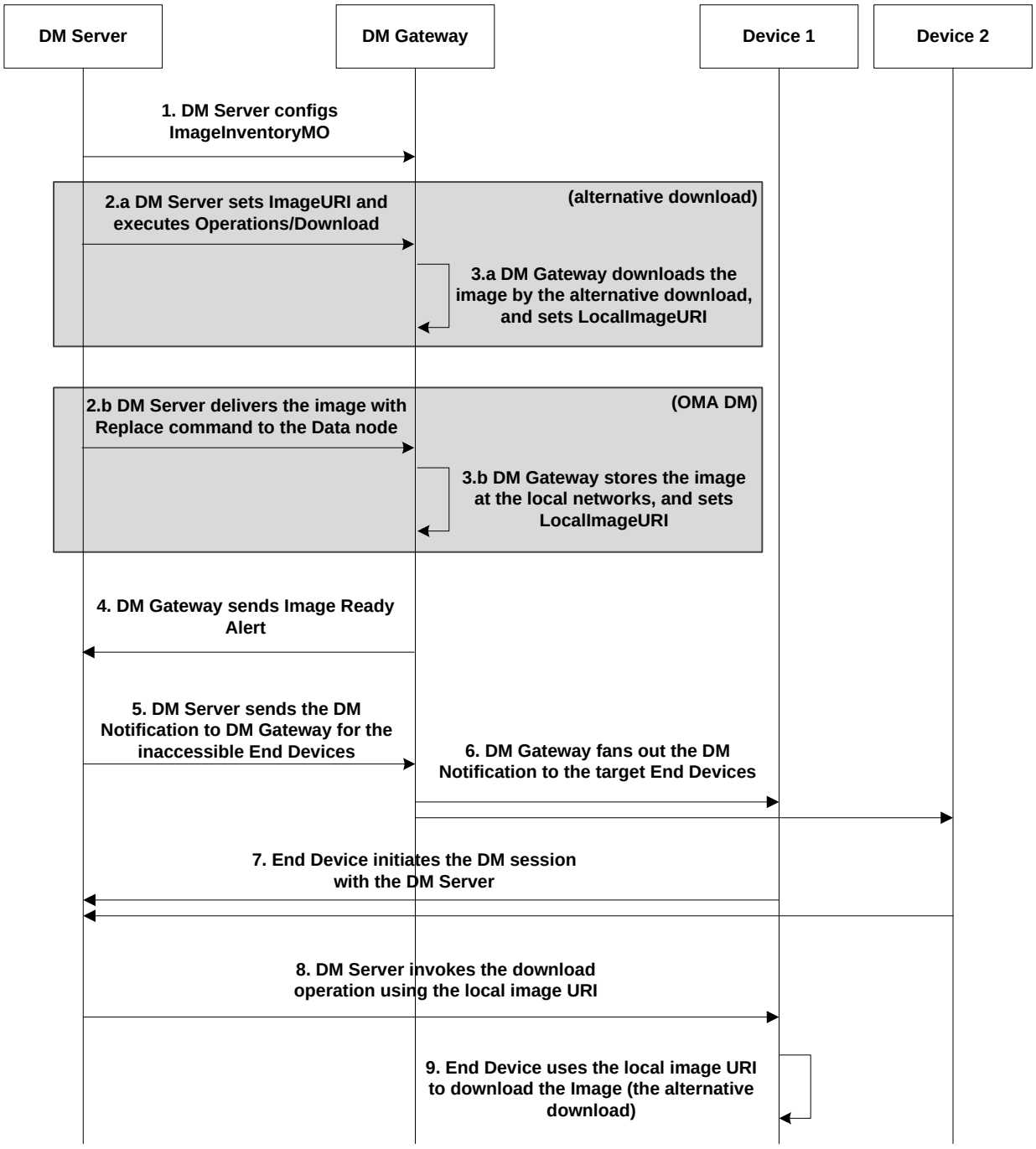

Figure 29: Image Distribution in Transparent Mode

Step 1 ~ Step 4: These steps are the same as Step 1 through Step 4 in section 8.6.1.

Step 5: The DM Server sends the DM Notification (Package #0) to the DM Gateway for the inaccessible End Devices. The GwMO Package #0 Push Message Header or the End Device Trigger MO can be used as described in the section 8.2.

Step 6: The DM Gateway forwards the DM Notification (Package #0) to the target End Devices.

Step 7: The End Device initiates the DM session with the DM Server.

Step 8: Within the context of the DM session, the DM Server invokes a download operation on the End Device, using the local image URI, provided by the DM Gateway previously in the Image Ready Alert.

Step 9: The End Device uses the local image URI to download the image.

Step 8 and Step 9 can be repeated for other End Devices.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-GwMO-V1_1-20170725-A	25 Jul 2017	Status changed to Approved by TP TP Ref # OMA-TP-2017-0031-INP_GwMO-V1_1_ERP_for_Final_Approval

Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for GwMO Tree Structure

Item	Function	Reference	Requirement
GwMO-T-001-M	Use of appropriate Management Object identifier for the GwMO node	Sections 6.1.1, 6.2.1, 6.3.1, 6.4.1	
GwMO-T-002-M	Support for Required nodes under the root node	Sections 6.1.1, 6.2.1, 6.3.1, 6.4.1	
GwMO-T-003-O	Support for Optional nodes	Sections 6.1.1, 6.2.1, 6.3.1, 6.4.1	
GwMO-T-004-M	Support for Required nodes under an Optional node if the Optional node is supported	Sections 6.1.1, 6.2.1, 6.3.1, 6.4.1	

B.2 SCR for GwMO Client

B.2.1 SCR for Device Inventory MO and DM Notification

Item	Function	Reference	Requirement
GwMO-C-001-M	Support for updating Device Inventory MO when the GW becomes aware of a new Device	Section 6.1.1, 8.1.1	
GwMO-C-002-M	Support for updating Device Inventory MO when the GW becomes aware of a previously subtending Device that is no longer present in the network	Section 6.1.1	
GwMO-C-003-M	Support for Device Inventory Alert	Section 6.1.1, 7.1.1	
GwMO-C-004-M	Support for Device Detach Alert	Section 6.1.1, 6.2.1, 7.1.2	
GwMO-C-005-M	Support for the Gateway in Proxy mode	Section 6.1.1	
GwMO-C-006-M	Support for the Gateway in Transparent and Proxy mode	Section 6.1.1	
GwMO-C-007-O	Support for the Gateway in Adaptation mode	Section 6.1.1, 0	
GwMO-C-008-M	Support for management of End Devices with different types of End Device addresses	Section 6.1.1	

Item	Function	Reference	Requirement
GwMO-C-009-M	Transparent mode and support GwMO Package #0 Push Message Header Format processing	Section 8.2.2	
GwMO-C-110-O	Support for updating Device Inventory MO with Hierarchical Architecture, when the GW becomes aware of a new Device via one or more GWs in a tree branch, including the ComponentType of the new Device.	Section 8.1.2	
GwMO-C-111-O	Support for Device Inventory Alert with Hierarchical Architecture, including new fields HierarchicalArchitecturePath and ComponentType	Section 7.1.1, Appendix C.1	

B.2.2 SCR for Gateway Config MO

Item	Function	Reference	Requirement
GwMO-C-010-M	Support for PopulateGroup operation	Section 6.2.1	
GwMO-C-011-M	Support for enumerated based group membership	Section 6.2.1	
GwMO-C-012-M	Support for group membership based on device type	Section 6.2.1	
GwMO-C-013-O	Support for other conditions set by DM Server for group membership	Section 6.2.1	
GwMO-C-112-O	Support for secure mechanism for the DM Gateway to operate in proxy mode operation, including the Privileges sub-tree	Section 6.2.1, 8.3.4	

B.2.3 SCR for Fanout MO

Item	Function	Reference	Requirement
GwMO-C-014-M	Support for Fanout operation	Section 6.3.1, 8.3	
GwMO-C-015-M	Support for Fanout Result Aggregation Alert	Section 6.3.1, 7.2	
GwMO-C-016-M	Support for Fanout Result Complete Status Alert	Section 6.3.1, 7.2	
GwMO-C-017-O	Support synchronous result reporting	Section 8.1.2	
GwMO-C-018-O	Adaptation mode and support non-DM commands	Section 8.4.3	

Item	Function	Reference	Requirement
GwMO-C-019-M	Support Fanout notification	Section 8.2	
GwMO-C-020-M	Support for result filtering	Section 6.3.1, 8.3.1	
GwMO-C-021-O	Support for create group based on result filtering	Section 6.3.1	

B.2.4 SCR for Image Inventory MO

Item	Function	Reference	Requirement
GwMO-C-022-M	Support download for image (for example,, delivery package)	Section 6.4.1	
GwMO-C-023-M	Support for expire image inventory	Section 6.4.1	
GwMO-C-024-M	Support for assignment of LocalImageURI operation	Section 6.4.1	

B.3 SCR for GwMO Server

Item	Function	Reference	Requirement
GwMO-S-001-M	Support for the Gateway Management Object	Section 6	
GwMO-S-002-M	Transparent mode and support GwMO Package #0 Push Message Header Format	Section 8.2.1	

Appendix C. XML Schema for GwMO Alerts (Normative)

Note:- The following alerts do not contain any data and therefore no XML Schema is defined for them:

- Fanout Completion Status Alert
- Image Ready Alert

C.1 Device Inventory Alerts (Device Attach/Device Detach)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="DeviceInventory">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Device" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Device">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="DeviceID"/>
        <xs:element ref="Mode" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="Address" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="AddressType" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="ComponentType" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="HierarchicalArchitecturePath" minOccurs="0"
maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DeviceID" type="xs:string"/>
  <xs:element name="Address" type="xs:string"/>
  <xs:element name="AddressType" type="xs:string"/>
  <xs:element name="Mode" type="xs:unsignedInt" />
  <xs:element name="ComponentType" type="xs:unsignedInt"/>
  <xs:element name="HierarchicalArchitecturePath" type="xs:string"/>
</xs:schema>
```

C.2 Fanout Result Aggregation Alert

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="Node">
    <xs:complexType>
      <xs:sequence>
  <xs:element name="DeviceID" type="xs:string"/>
   <xs:element name="DevResult" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

C.3 Bootstrapped DMS List Alert

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="ServerList">
    <xs:complexType>
      <xs:element ref="ServerInfo" maxOccurs="unbounded">
    </xs:complexType>
  </xs:element>
  <xs:element name="ServerInfo">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ServerID" type="xs:string"/>
        <xs:element name="BootstrapURL" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

C.4 Associated Gateway Alert

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="GwAddress" type="xs:string"/>
</xs:schema>
```