ETSI TR 118 506 V1.0.0 (2015-04)

TECHNICAL REPORT

Study of Management Capability Enablement Technologies for Consideration by oneM2M

Reference

DTR/oneM2M-000006

Keywords

architecture, IoT, M2M, management,
requirements

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Report (TR) has been produced by ETSI Partnership Project oneM2M (oneM2M).

1 Scope

The present document describes and collects the state-of-art of the existing technologies on management capability, evaluates if the technologies can match the requirements defined in oneM2M, analyzes how the technologies can leverage the design of the architecture of oneM2M.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

Not applicable.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] oneM2M Drafting Rules.

NOTE: Available at ftp://ftp.onem2m.org/Others/Rules_Pages/oneM2M-Drafting-Rules-V1_0.doc.

[i.2] OMA-AD-DM-V1_3: "Device Management Architecture".

[i.3] OMA-TS-DM-Protocol-V1_3: "OMA Device Management Protocol".

[i.4] OMA-TS-DM-RepPro-V1_3: "OMA Device Management Representation Protocol".

[i.5] OMA-TS-DM-StdObj-V1_3: "OMA Device Management Standardized Objects".

[i.6] OMA-TS-DCMO-V1_0: "Device Capability Management Object".

[i.7] OMA-TS-LAWMO-V1_0: "Lock and Wipe Management Object".

[i.8] OMA-TS-DM-FUMO-V1_0: "Firmware Update Management Object".

[i.9] OMA-TS-DM-SCOMO-V1_0: "Software Component Management Object", Version 1.0.

[i.10] OMA-TS-GwMO-V1_0: "Gateway Management Object Technical Specification", Version 1.0.

[i.11] OMA-TS-DiagMonFunctions-1_0: "DiagMon Functions Supplemental Specification", Version 1.0.

[i.12] OMA-AD-GwMO-V1_1-20130214-D: "Gateway Management Object Architecture".

[i.13] BBF TR-069: "CPE WAN Management Protocol", Issue: 1 Amendment 4, July 2011.

[i.14] BBF MR-239: "Broadband Forum Value Proposition for Connected Home", Issue: 1, April 2011.

[i.15] BBF TR-232: "Bulk Data Collection", Issue: 1, May 2012.

[i.16] TMForum IPDR Service Specification Design Guide, Version 3.8, Release 1.0, 2009.

[i.17] OMA-RD-LightweightM2M-V1_0: "OMA Lightweight Machine to Machine Requirement".

[i.18] OMA-AD-LightweightM2M-V1_0: "OMA Lightweight Machine to Machine Architecture".

[i.19] OMA-TS-LightweightM2M-V1_0: "OMA Lightweight Machine to Machine Protocol" (work on progress).

[i.20] draft-ietf-core-coap-14 (work in progress), Sept 2012: "Constrained Application Protocol (CoAP)", Z. Shelby, K. Hartke, C. Bormann and B. Frank.

[i.21] IETF RFC 6347 (January 2012): "Datagram Transport Layer Security Version 1.2", E. Rescorla and N. Modadugu.

[i.22] oneM2M-TS-0002 (V0.5.2): "oneM2M Requirements".

[i.23] ETSI TS 103 092 (V2.1.1): "Machine-to-Machine communications (M2M); OMA DM compatible Management Objects for ETSI M2M".

[i.24] OMA-ER-DMClientAPIfw-V1_0: "DM Client Side API Framework (DMClientAPIfw)".

[i.25] IETF draft-ietf-lwig-terminology-05 (July 09 2012): "Terminology for Constrained Node Networks", C. Bormann and M. Ersue.

[i.26] IEEE Communication Magazine (Vol.50, Issue.12) (Dec 2012): "Management of Resource Constrained Devices in the Internet of Things", A. Sehgal, V. Perelman, S Kuryla and J Schonwalder.

[i.27] BBF TR-131: "ACS Northbound Interface Requirements", Issue:1, November 2009.

[i.28] BBF TR-143: "Enabling Network Throughput Performance Tests and Statistical Monitoring", Corrigendum 1, December 2008.

[i.29] BBF TR-181: "Device Data Model for TR-069", Issue 2 Amendment 6, November 2012.

[i.30] BBF TR-135: "Device Data Model for TR-069 Enabled STB", Amendment 3, November 2012.

[i.31] BBF TR-104: "Provisioning Parameters for VoIP CPE", September 2005.

[i.32] BBF TR-140: "TR-069 Data Model for Storage Service Enabled Devices", Issue 1.1, December 2007.

[i.33] OMA-TS-DM_Security-V1_2_1: "OMA Device Management Security".

[i.34] oneM2M TS-0001: "oneM2M Functional Architecture".

[i.35] IETF RFC 2616: "Hypertext Transfer Protocol -- HTTP/1.1".

[i.36] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions.apply:

mc: interface between the management server and the management client

NOTE: This interface can be realized by the existing device management technologies such as BBF TR-069 [i.13], OMA DM, etc.

ms: interface between the management adapter and the management server in the underlying network domain or in the M2M service domain for use by other systems

NOTE: Using this interface, systems can perform management operations on devices through the management server.

mp: interface that is exposed by the proxy management client in the area network for devices that connect to a proxy

NOTE: This interface is realized by existing LAN based protocols (e.g. ZigBee®, UPnP) as well as existing device management technologies (e.g. OMA-DM).

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ACL	Access Control List
ACS	Auto-Configuration Server
API	Application Programming Interface
ASN	Application Service Node
ATM	Asynchronous Transfer Mode
BBF	Broadband Forum
BSS	Business Support System
CoAP	Constrained Application Protocol
CP	Client Provisioning
CPE	Customer Premises Equipment
CPU	Centralized Processing Unit
CSF	Common Services Function
CWMP	CPE WAN Management Protocol
DCMO	Device Capability Management Object
DM	Device Management
DNS	Domain Name Server
DRAM	Dynamic Random-Access Memory
DSL	Digital Subscriber Line
DTD	Document Type Definition
DTLS	Datagram Transport Layer Security
FTP	File Transfer Protocol
FUMO	Firmware Update Management Object
GPS	Global Positioning System
GSM	Global System for Mobile communications
GW	Gateway
HPNA	Home Phoneline Networking Alliance
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IO	Input Output
IP	Internet Protocol
IPDR	Internet Protocol Detail Record
IPSO	IP for Smart Objects
IrDA	Infrared Data Association
JSON	JavaScript Object Notation
KB	KiloBytes
LAN	Local Area Network

LAWMO	Lock and Wipe Management Object
MANDMO	M2M Area Network Device Management Object
MANMO	M2M Area Network Management Object
MCU	MicroController Unit
MGR	Management Requirement
MIID	MO Instance Identifier
MMS	Multimedia Messaging Service
MO	Management Object
MOID	Management Object Identifier
MR	Marketing Report
NAT	Network Address Translation
NS	Name Server
OBEX	OBject EXchange
OMA	Open Mobile Alliance
OS	Operating System
OSS	Operation Support System
OTA	Over The Air
RAM	Random Access Memory
RFC	Request For Comment
RPC	Remote Procedure Call
SCOMO	Software Component Management Object
SCTP	Stream Control Transmission Protocol
SE	Service Element
SIP	Session Initiation Protocol
SMS	Short Message Service
SOAP	Simple Object Access Protocol
SSL	Secure Session Layer
STB	Set Top Box
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol and the Internet Protocol
TLS	Transport Layer Security
TLV	Type-Length-Value
TMForum	Telemanagement Forum
TR	Technical Report
UDP	User Datagram Protocol
UI	User Interaction
UPA	Universal Powerline Association
UPnP DM	Universial Plug and Play Device Management
URI	Uniform Resource Identifier
USB	Universal Serial Bus
WAN	Wide Area Network
WCDMA	Wideband Code Division Multiple Access
WLAN	Wireless Local Area Network
WSP	Wireless Session Protocol
XDR	External Data Representation
XML	Extensible Markup Language
ZC	ZigBee® Coordinator
ZDO	ZigBee® Device Object

4 Conventions

The key words "Shall", "Shall not", "May", "Need not", "Should", "Should not" in this document are to be interpreted as described in the oneM2M Drafting Rules [i.1].

5 Introduction of existing technologies

5.1 Introduction to OMA DM

5.1.1 Description

OMA DM is a protocol for device management designed by Open Mobile Alliance. It is widely used in the remote management of mobile devices. It is composed of a number of specifications including protocol, architecture, underlying network binding etc. In the most common scenario, by implementing OMA DM specifications, the DM Server is able to do remote management on devices with DM Clients which are usually mobile phones. The devices could also include sensors, actuators, and gateways as well. With implementing the Management Object and the DM Client, the DM Server can perform remote management on devices such as provisioning, diagnostics, firmware upgrade, and remove, install, activate software components.

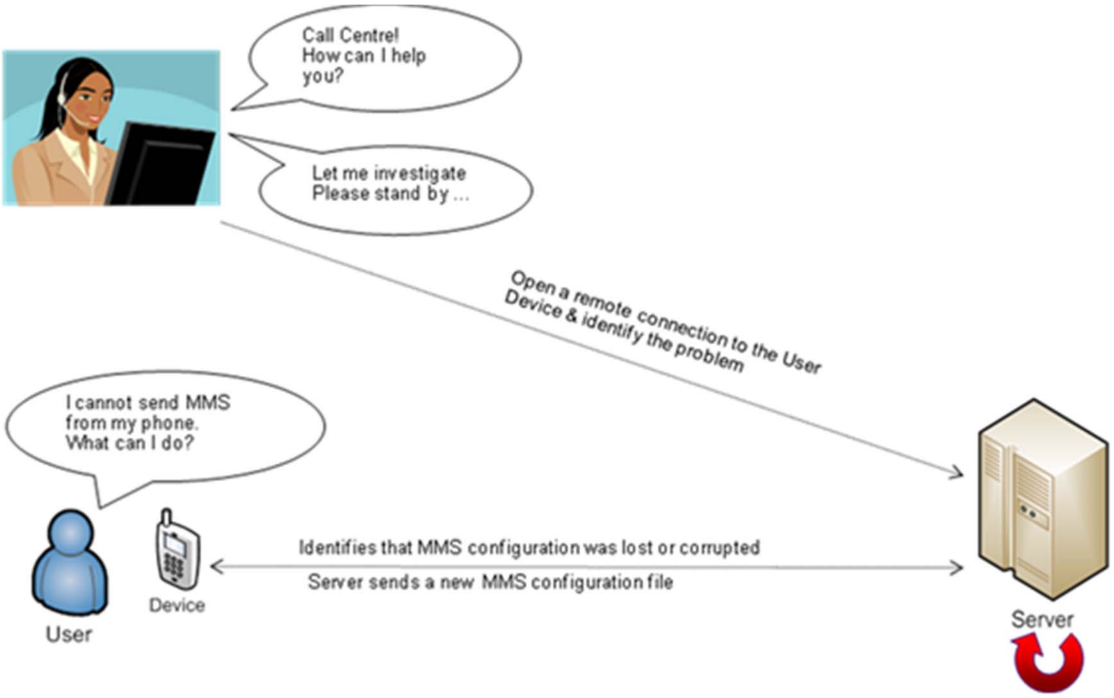

Call Centre!
How can I help you?
Let me investigate
Please stand by ...
Open a remote connection to the User Device & identify the problem
I cannot send MMS from my phone. What can I do?
User
Device
Identifies that MMS configuration was lost or corrupted
Server sends a new MMS configuration file
Server

Figure 5.1.1: OMA DM Use Case

As is shown from figure 5.1.1, the user of a mobile phone does not know what to do when his mobile is unable to send out MMS. After calling to the Call Center, the operator of the Call Center can remotely upgrade the MMS configuration file via OMA DM Server.

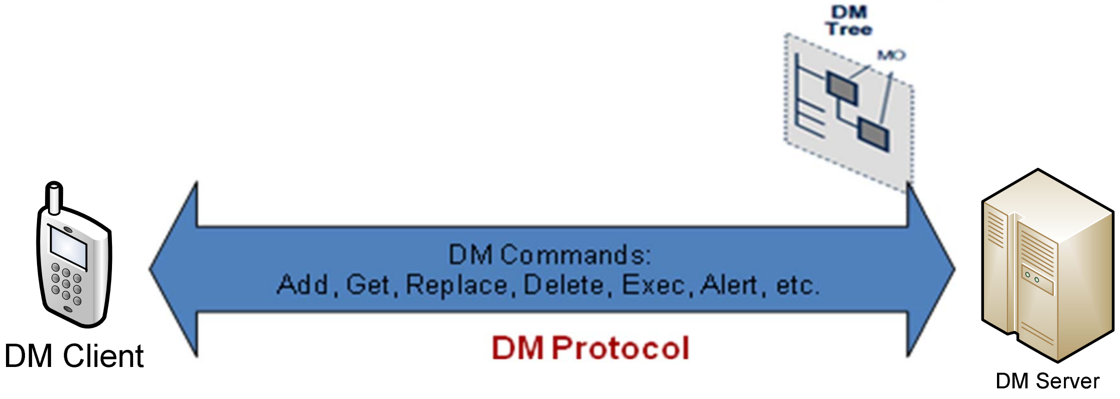


Figure 5.1.2: Device Management Protocol

OMA DM protocol deploys management between a DM Client and a DM Server as figure 5.1.2. DM Server can send DM commands to DM Client to manage the device. The DM Client can also send command to DM Server to indicate the result corresponding to the commands from DM Server. A group of tree structured Management Objects are used to manage the device.

5.1.2 Architecture

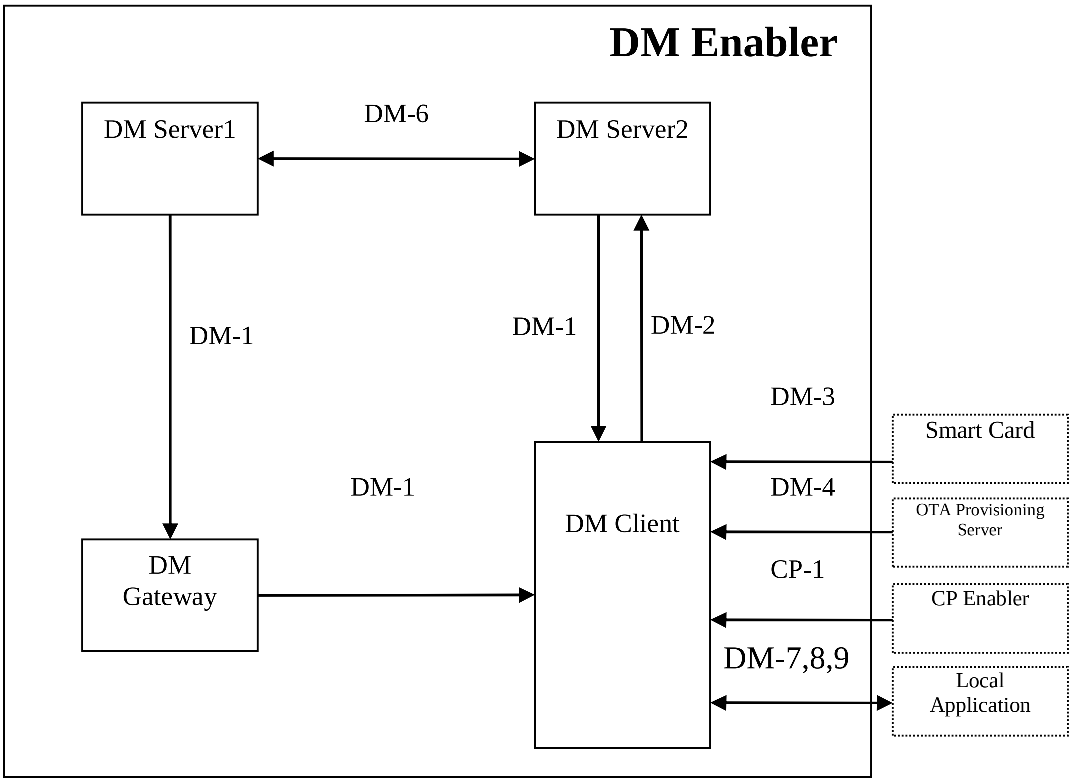


Figure 5.1.3: Architecture and Reference Points

The architecture of OMA Device Management Enabler [i.2] is shown in figure 5.1.3. Functional components which are DM Server and DM Client compose the DM Enabler. Components Smart Card, OTA Provisioning Server and CP Enabler are outside of the DM Enabler. They are used to bootstrap the DM Client.

DM Server can also manage a device with DM Client through a DM Gateway. DM Gateway can be deployed in DM-1 interface in Transparent Mode, Proxy Mode or Adaption Mode [i.12].

5.1.3 Reference points

5.1.3.1 Introduction

This clause introduces the interfaces carried over the reference points between DM Server, DM Client, Smart Card, OTA Provision Server and CP Enabler. Also the procedures of packages exchanged via these interfaces are also briefly introduced.

5.1.3.2 DM-1 DM Client-Server Notification

The DM-1 interface provides the ability for the DM Servers to send device management notifications to the DM Clients. Because devices with DM Clients may not be able to continuously listen for connection all the time, DM Server may send notifications to DM Client to start a DM session. More details can be referred to [i.2].

5.1.3.3 DM-2 DM Client-Server Protocol

The interface provides the ability for the DM Servers and DM Clients to exchange DM commands and corresponding responses. The interface can be bound to different underlying protocols including HTTP and HTTPS. More details can be referred to [i.2].

5.1.3.4 DM-3 DM Bootstrap Profile via Smart Card

Bootstrap via Smart Card is one way to provision a DM Client. The DM Client gets all the related configuration settings from the Smart Card. More details can be referred to [i.2].

5.1.3.5 DM-4 DM Bootstrap Profile OTA

Bootstrap via push protocol over the air can provision necessary configuration setting file to DM Client. The file contains a series of DM Commands. More details can be referred to [i.2].

5.1.3.6 CP-1 CP Bootstrap Profile

Bootstrap via CP enabler can provision necessary configuration setting file to DM Client. The file contains a series of DM Commands. More details can be referred to [i.2].

5.1.3.7 DM-6 DM Server-Server Interface

DM Server-Server Interface enables one DM Server delegate the management of a device to another DM Server. More details can be referred to [i.2].

5.1.3.8 DM-7,8,9 Client API

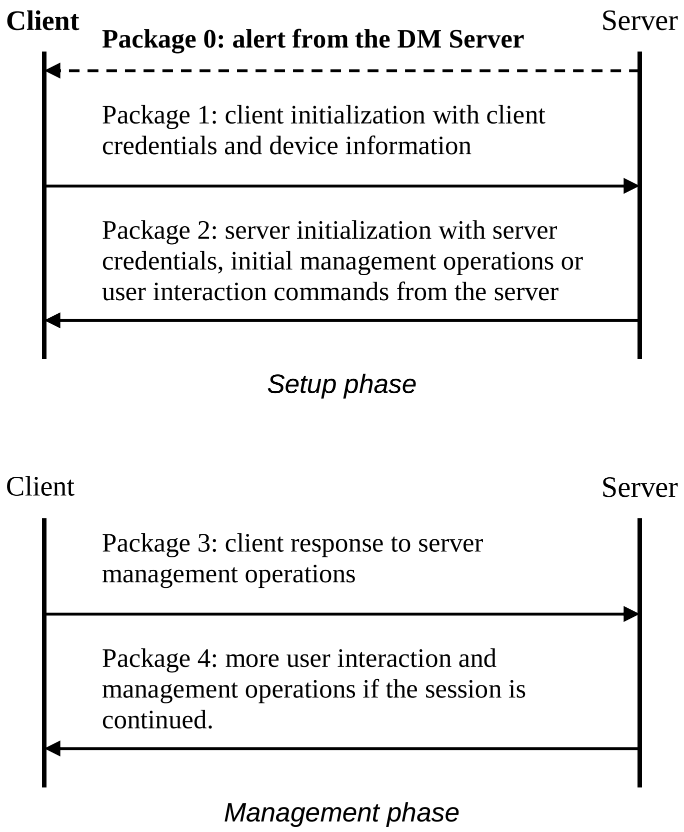
DM-7 is the interface that enables the local application of a device to register or deregister Management Object to the DM Client. [i.24].

DM-8 is the interface that enables the DM Client to send Management Object update notifications to local application. [i.24].

DM-9 is the interface that enables the local application to send Management Object manipulation and retrieve commands to the DM Client. [i.24].

Local application resides in the same execution environment with the DM Client.

5.1.3.9 Procedures


Client
Package 0: alert from the DM Server
Server
Package 1: client initialization with client credentials and device information
Package 2: server initialization with server credentials, initial management operations or user interaction commands from the server
Client
Server
Package 3: client response to server management operations
Package 4: more user interaction and management operations if the session is continued.

Setup phase

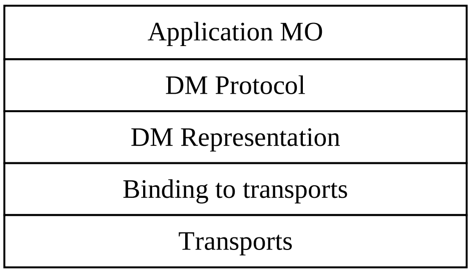
Management phase

Figure 5.1.4: DM Phases

The interaction between Client and Server is achieved by Packages. OMA DM Protocol [i.3] consists of two parts: setup phase (authentication and device information exchange) and management phase. Management phase can be repeated as many times as the DM Server wishes. The setup phase is composed of Pachage#0, Package#1 and Package#2. The management phase is composed of Package#3 and Package#4 as shown in figure 5.1.4.

5.1.4 Protocols

5.1.4.1 Protocol Stack


Application MO

DM Protocol

DM Representation

Binding to transports

Transports

Figure 5.1.5: Protocol Stack

As shown in figure 5.1.5, the protocol stack of OMA DM is composed of five layers which are Application MO, DM Protocol, DM Representation, Binding to transports and Transports.

5.1.4.2 Application MO

The Management Object is built on top of the DM Protocol to be transferred to fulfil the management of devices. MO is implemented in the device with DM Client and DM Server to carry on the management. DM Server manages the device by operation to the MO through DM Client. The introduction to the MOs will be shown in clause 5.1.5.

5.1.4.3 DM Protocol

DM Protocol is the Packages exchanged between the entities of OMA DM. As is described in the reference point part, OMA DM uses these Packages to exchange the MO between DM Client and DM Server.

5.1.4.4 DM Representation

OMA DM uses DM representation syntax and semantics for device management. The DM representation is carried in the XML formatted DM Messages between OMA DM entities. The DM representation protocol also can be identified as a MIME content type.

The DM representation protocol is performed in a request/response way using the concept of DM Package. The concept of DM Package is shown in the Procedure chapter. It's used to carry the device management operations.

A DM Message is a well-formed XML document and adheres to the DTD.

OMA DM uses SyncML as the container for the DM Message. SyncML was first designed and used by OMA CP, and was reused by OMA DM. SyncML provides a set of tags and syntaxes to mark up the language to be understandable to both DM Clients and DM Servers.

Details can be referred to [i.4].

5.1.4.5 Binding to transports and Transports

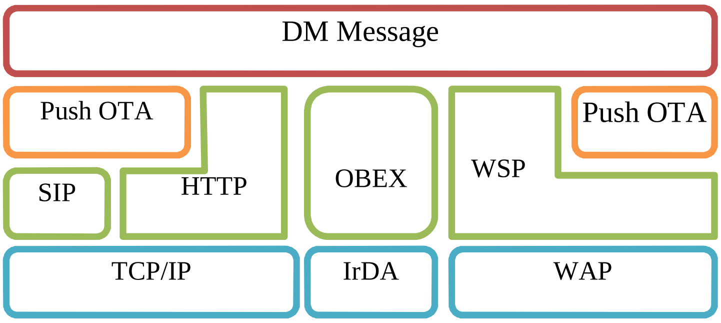


Figure 5.1.6: OMA DM Transports

OMA DM provides the following ways for transporting DM Messages which are HTTP, OBEX, WSP, SIP and Push OTA as shown in figure 5.1.6 OMA DM Transports.

5.1.5 Functions

5.1.5.1 Introduction

The device management functionalities are achieved by the Management Objects defined by OMA DM and some other third party organizations.

5.1.5.2 The MO tree

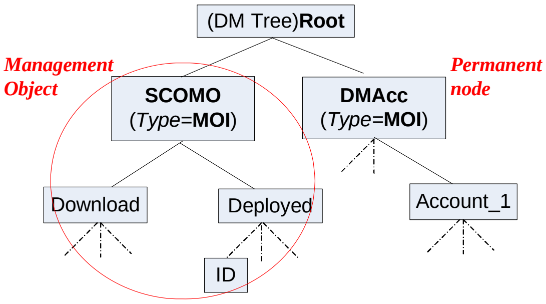


Figure 5.1.7: MO tree

OMA DM uses Management Object to manage the device. The MOs forms a tree structure and the tree is stored with the DM Client. Each MO in the tree is a node. If the MO has child Nodes, the MO is an Interior Node. Otherwise, the MO is a Leaf Node. Nodes in the Management Tree can be either permanent or dynamic.

Permanent Nodes are typically built in at device manufacture. Permanent Nodes can also be temporarily added to a device by, for instance, connecting new accessory hardware. A DM Server cannot modify permanent Nodes at run-time.

Dynamic Nodes can be created and deleted at run-time by DM Servers. DM Server use Add and Delete command to create or delete Dynamic Nodes. If the deleted Dynamic Nodes is an Interior Node, all the related Nodes which are the children of the Interior Node shall also be deleted.

5.1.5.2.1 Standard Objects

The MOs that shall be supported by DM Client and DM Server are standard objects. Standard objects expose basic information of the DM Client for the DM Server to perform managements.

Table 5.1.1: Standard Objects

Management Object	Reference	Description
DMAcc	[i.5]	Settings for the DM client in a managed device.
DevInfo	[i.5]	Device information for the OMA DM server. Sent from the client to the server. Needed by the DM Server for problem free operation of the DM protocol.
DevDetail	[i.5]	General device information that benefits from standardization. DevDetail contains parameters that are manipulated by the server for the operation purposes.
Inbox	[i.5]	Reserved URI where the device uses the management object identifier to identify the absolute URI.

5.1.5.2.2 Other Management Objects

Besides the Standard Objects, there may be other Management Objects to carry on further management functionalities as well. MOs that are considered as relevant to the management of M2M Devices or Gateways are listed in table 5.1.2.

Table 5.1.2: Other MOs

Management Object	Reference	Description
SCOMO	[i.9]	Device information collection, remote configuration, software management
DIAGMON	[i.11]	Diagnostics and monitoring
GwMO	[i.10]	Managements to devices through gateway [i.5]
FUMO	[i.8]	Firmware update
DCMO	[i.6]	Specify the mechanisms required for the remote management of device capabilities
LAWMO	[i.7]	The MO is designed to protect user and enterprise-related data by means including Lock/Unlock Device, Wipe Device's Data and Factory Reset

5.2 TR-069 Family of Specifications

5.2.1 Description

The Broadband Forum has developed a series of specifications that have been termed the TR-069 Family of Specifications. These specifications provides the capability to manage CPEs within the connected home. MR-239 Broadband Forum Value Proposition for Connected Home [i.14] provides an overview of the value proposition for utilizing the TR-069 family of specifications for the connected home.

5.2.2 Architecture

The TR-069 family of specifications is anchored by the TR-069 [i.13] specification for the CPE WAN Management Protocol (CWMP) protocol. TR-143 [i.28] for diagnostic tests. TR-131 [i.27] for requirements related to the ACS North Bound Interface.

In addition Service Providers are increasingly interested in retrieving large quantities of data from their installed CPE base at regular intervals. The amount of data being requested represents a significant portion of the CPE's data model and is thus a large amount of data. In response to this, the Broadband Forum has documented a data collection solution in TR-232 [i.15] Bulk Data Collection. This specification is based on the IPDR protocol from the TMForum.

Devices within the Connected Home are managed via a set of data models for CWMP Enabled Devices. These data models are anchored by TR-181 [i.29] which defines the objects and attributes for management for most capabilities offered by a device (e.g. physical interfaces, bridging and routing, firewalls, NAT, software modules). Likewise services and capabilities specific to a type of device are included in a separate set of specifications. For example, CWMP enabled STB are managed using TR-135 [i.30]; Femto cell devices are managed using TR-196 [i.32]; VoIP capable devices are managed using TR-104 [i.31]. Not all devices within the Connected Home are CWMP enabled; in this situation TR-069 [i.13] provides the capability for a CWMP enabled device to act a proxy for the device.

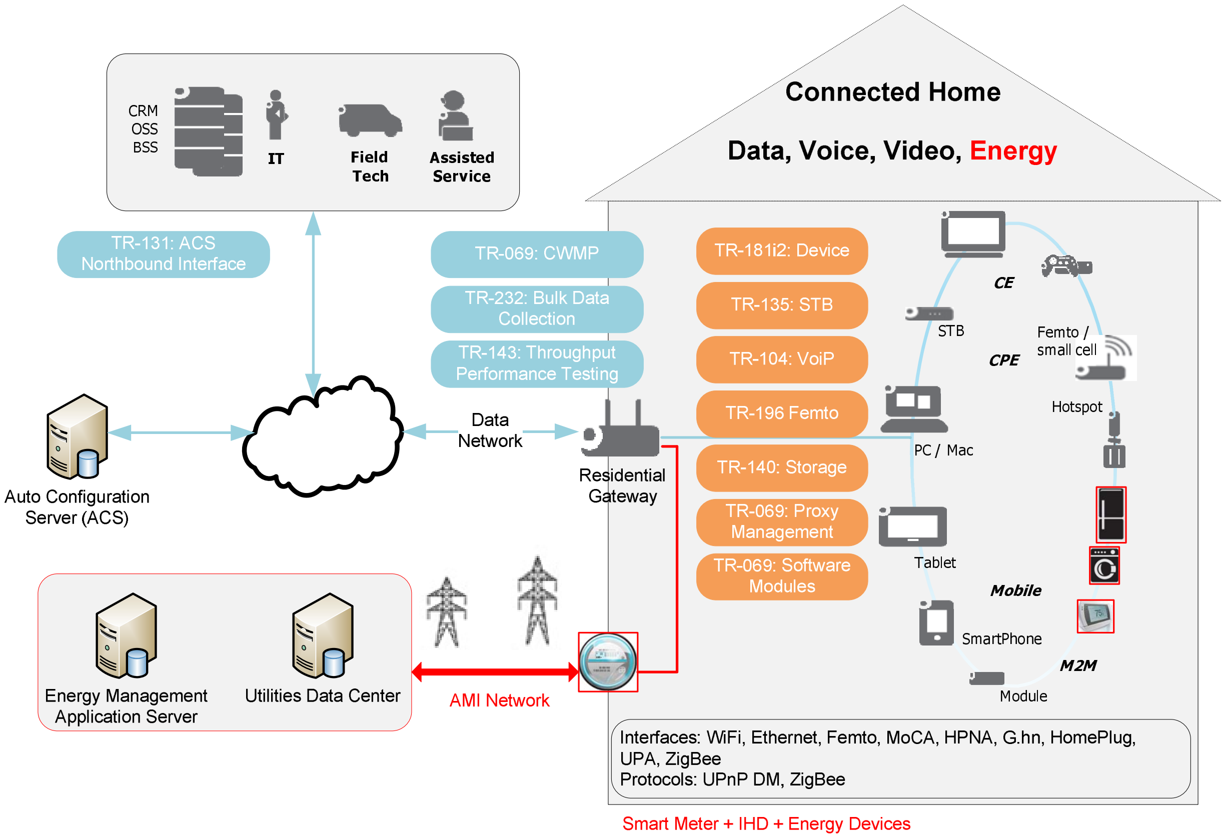

CRM
OSS
BSS
IT
Field Tech
Assisted Service
TR-131: ACS Northbound Interface
TR-069: CWMP
TR-232: Bulk Data Collection
TR-143: Throughput Performance Testing
Auto Configuration Server (ACS)
Data Network
Residential Gateway
Energy Management Application Server
Utilities Data Center
AMI Network
Connected Home
Data, Voice, Video, Energy
TR-181i2: Device
TR-135: STB
TR-104: VoiP
TR-196 Femto
TR-140: Storage
TR-069: Proxy Management
TR-069: Software Modules
CE
STB
CPE
Femto / small cell
Hotspot
PC / Mac
Tablet
Mobile
SmartPhone
M2M
Module
Interfaces: WiFi, Ethernet, Femto, MoCA, HPNA, G.hn, HomePlug, UPA, ZigBee
Protocols: UPnP DM, ZigBee
Smart Meter + IHD + Energy Devices

Figure 5.2.1: TR-069 [i.13] Family of Specifications

5.2.2.1 TR-069 Proxy Management

CWMP can be extended to devices that do not have a native CWMP Endpoint of their own, but instead support management of devices with another management protocol or "Proxy Protocol". A CPE Proxier is a CPE that supports a CWMP Endpoint(s) and also supports one or more Proxy Protocols (example services include UPnP DM, Z-Wave, etc.). A CPE Proxier uses these Proxy Protocols to manage the devices connected to it, i.e. the Proxied Devices. This approach is designed to support Proxy Protocols of all types that can exist in the CPE network now or in the future. Annex J of the TR-069 [i.13] provides an overview of CWMP Proxy Management.

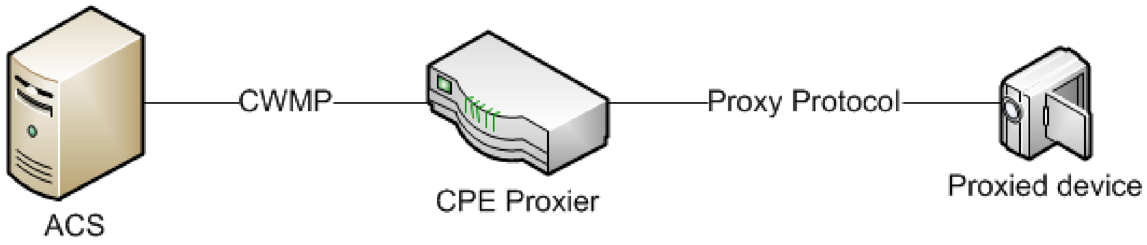


Figure 5.2.2: Proxy management terminology

5.2.2.1.1 Proxied Device Deployment Archirecture

Figure 5.2.3: TR-069 [i.13] UPnP DM Proxied Device depicts an example scenario where a proxied device that supports the UPnP DM protocol is managed using CWMP.

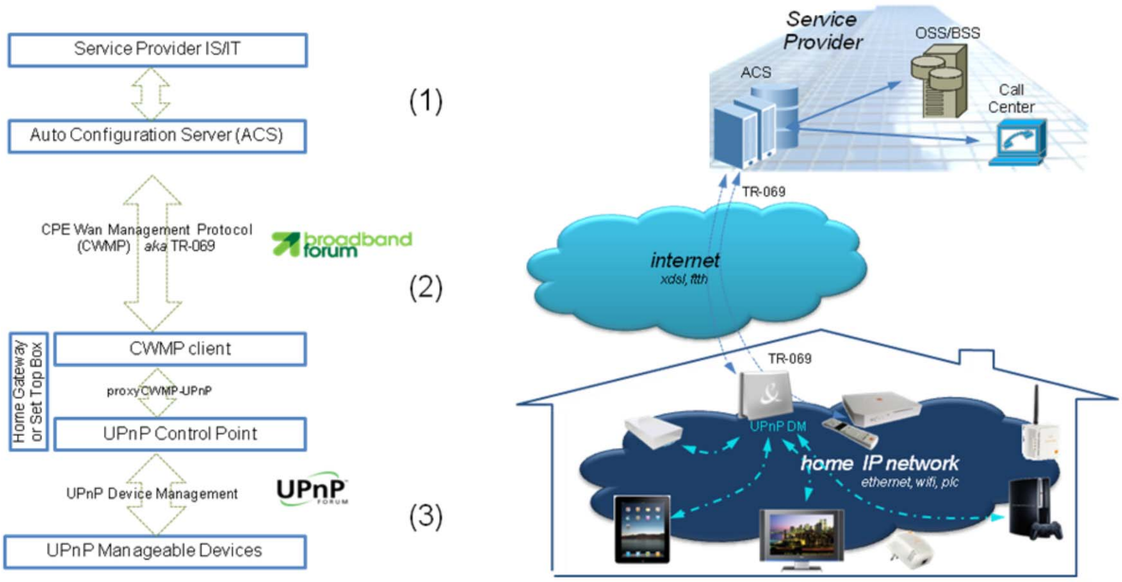


Figure 5.2.3: TR-069 [i.13] UPnP DM Proxied Device

The entities include the Service Provider OSS/BSS systems that interface with the ACS (1); the CWMP and IPDR protocols between the ACS and the TR-069 [i.13] enabled CPE (2) and the Home Area Network protocol UPnP (3).

5.2.3 Reference points

The CWMP enabled devices in the Connected Home typically communicates with three (3) entities, the ACS, OSS/BSS and devices within the Connected Home via standardized reference points.

These references points are defined as:

- ACS to CPE.
- CPE to BSS.
- CPE to Device.

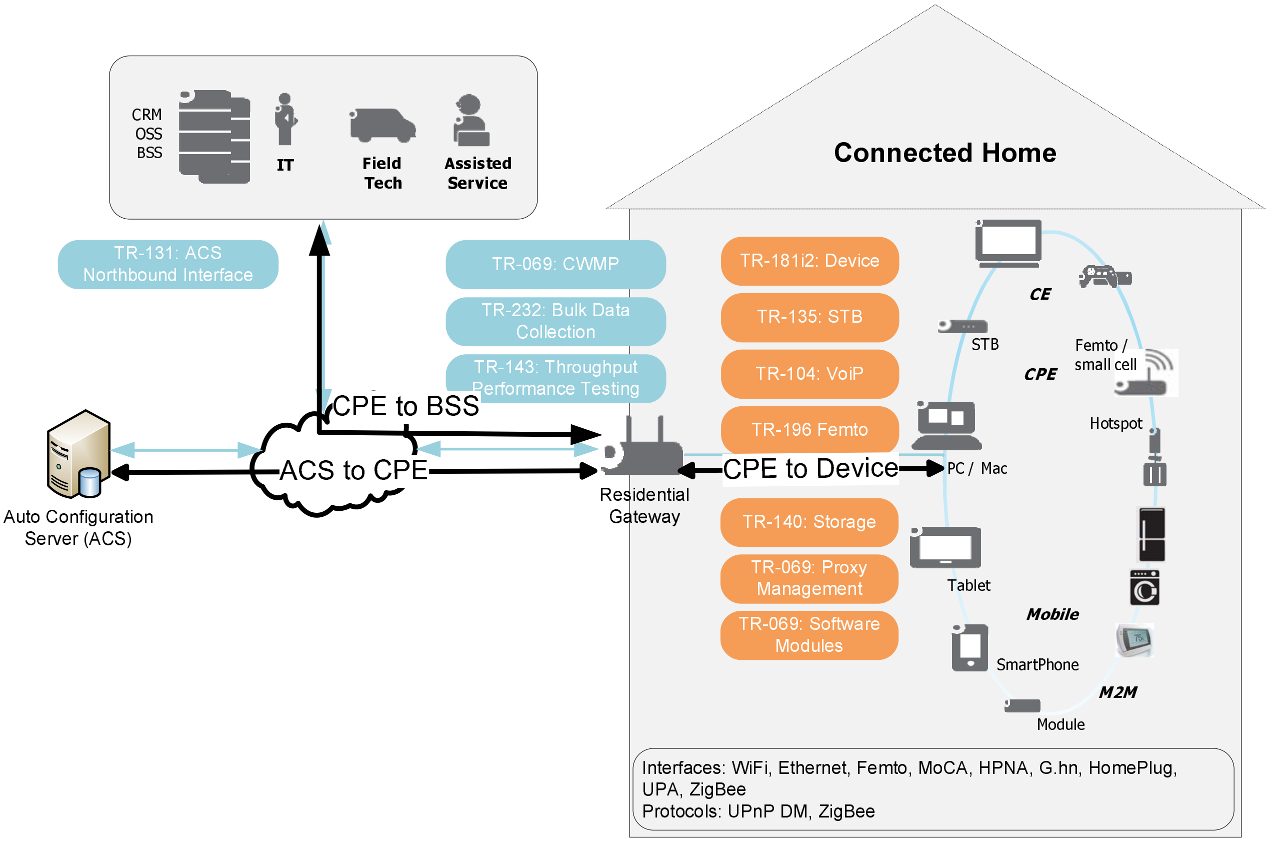
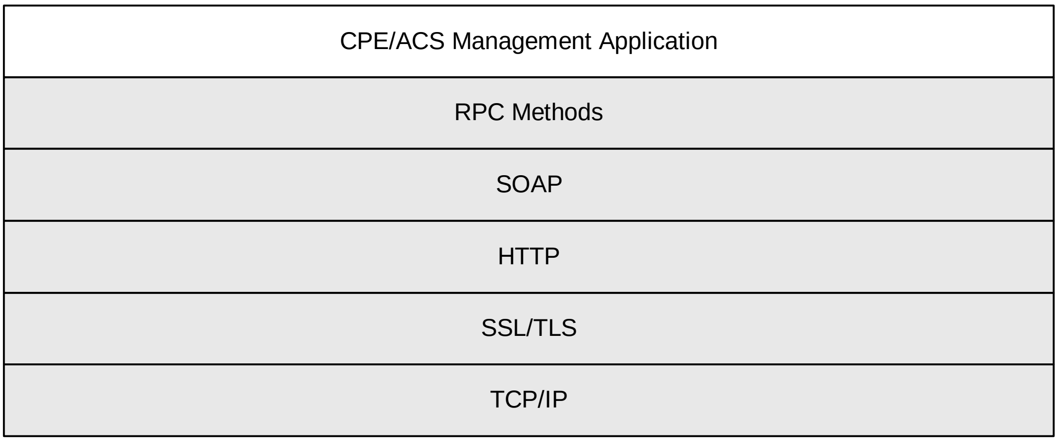


Figure 5.2.4: TR-069 Reference Points

5.2.4 Protocols

5.2.4.1 ACS to CPE Protocol

The protocol that is supported on the ACS to CPE reference point is CWMP as defined in BBF TR-069 [i.13]. CWMP takes a layered approach to the protocol based on several standard protocols for transport and exchange of messages. The protocol stack defined byCWMP is shown in figure 5.2.5. A brief description of each layer is provided in table 5.2.1.



CPE/ACS Management Application

RPC Methods

SOAP

HTTP

SSL/TLS

TCP/IP

Figure 5.2.5: Protocol stack

Table 5.2.1: Protocol layer summary

Layer	Description
CPE/ACS Application	The application uses the CPE WAN Management Protocol on the CPE and ACS, respectively. The application is locally defined and not specified as part of the CPE WAN Management Protocol.
RPC Methods	The specific RPC methods that are defined by the CPE WAN Management Protocol. These methods are specified in CPE WAN Management Protocol.
SOAP	A standard XML-based syntax used here to encode remote procedure calls. Specifically SOAP 1.1, as specified in Simple Object Access Protocol (SOAP) 1.1.
HTTP	HTTP 1.1, as specified in IETF RFC 2616 [i.35], Hypertext Transfer Protocol -- HTTP/1.
TLS	The standard Internet transport layer security protocol. Specifically, TLS 1.2 (Transport Layer Security) as defined in IETF RFC 5246 [i.36], The Transport Layer Security (TLS) Protocol, Version 1.2 (or a later version). Note that previous versions of this specification referenced SSL 3.0 and TLS 1.0.
TCP/IP	Standard TCP/IP.

5.2.4.2 CPE to BSS

The protocol that is supported on the CPE to BSS reference point is the IPDR protocol. The IPDR reference architecture is presented in figure 5.2.6 is defined in TMForum IPDR Service Specification Design Guide [i.16]. The figure depicts a Service Element communicating to an IPDR Recorder that sends messages to the IPDR Transmitter and optionally to an IPDR Store. TR-232 [i.15] utilizes the A and D interfaces of this specification where the Service Element is a device within the Connected Home.

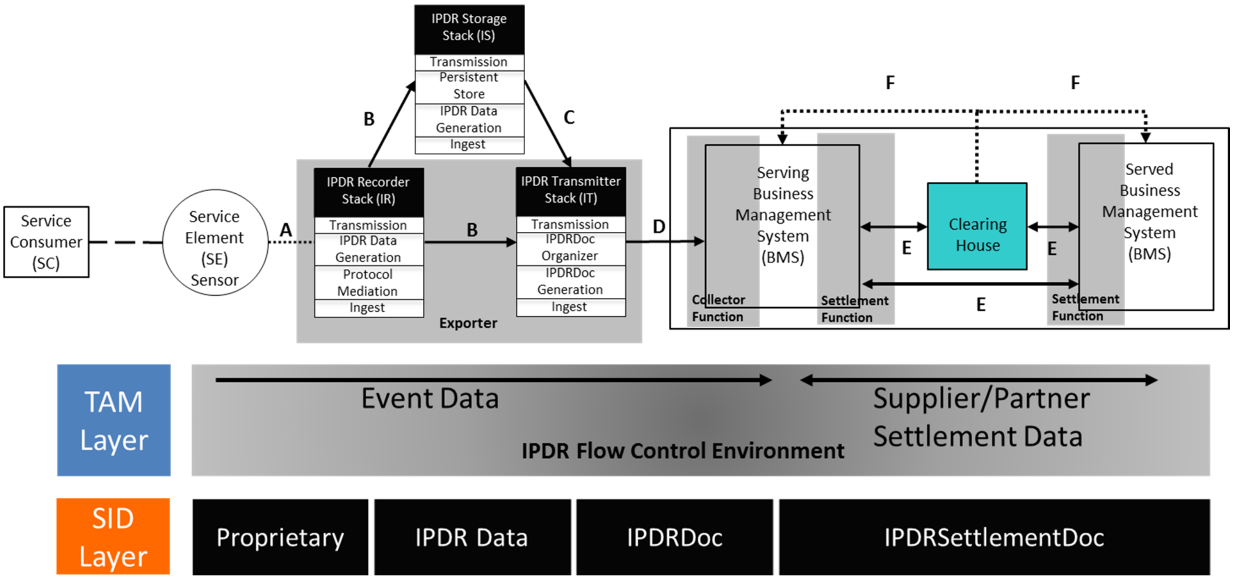


Figure 5.2.6: IPDR Reference Architecture

From the perspective of the Broadband Forum, the CPE or device is the Service Element and IPDR Exporter. The IPDR Data Collector is the BSS. As described in Annex A IPDR Theory of Operaton of TR-232 [i.15], the IPDR documentation clarifies that the following scenario, where the Service Element directly communicates to the BSS, is valid and simply means that the IPDR Recorder and IPDR Transmitter (collectively the IPDR Exporter in this use case) are all incorporated into the Service Element. The Service Element is permitted to directly interface with the BSS if it supports the "D" interface specifications including backing stores and retransmission of IPDR documents.

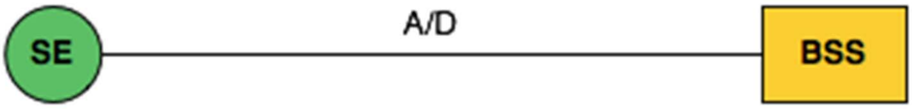


Figure 5.2.7: Simplified IPDR Architecture

5.2.4.2.1 IPDR Reference Points

TR-232 [i.15] defines 6 interfaces and 4 defintions for the IPDR Reference Model.

Table 5.2.2: IPDR Interfaces

Interface	Description
A	Vendor proprietary. High-volume with high granularity void of context. **This interface is not part of the IPDR Protocol.**
B	IPDR Data Interface. From IPDR Recorders to IPDR Stores or IPDR Transmitters.
C	IPDR Store Export Interface.
D	BSS Interface. XML or XDR data from IPDR Exporter to IPDR Collector
E	Settlement Interface. Connects Service Delivery Business Management Systems.
F	Financial System Interface. **This interface is not part of the IPDR Protocol.**

The IPDR File Transfer Protocol uses FTP or HTTP to transfer files that contain IPDR records from the SE to the BSS. The IPDR Streaming Protocol uses SCTP or TCP to transfer IPDR records from the SE to the BSS using highly efficient XDR encoding as described in the IPDR/XDR Encoding Format document or an XML encoding as described in the IPDR/XML File Encoding Format document.

5.2.4.3 CPE to Device Protocol

The TR-069 [i.13] proxy mechanism is designed to incorporate any protocol for area networks within the customer premises. The following protocols have been standardized or are currently in development:

- UPnP DM.
- ZigBee®.

5.2.4.3.1 UPnP DM Proxy

The CPE Proxier consists of three logical modules: CWMP client, TR-069/UPnP DM Proxy Module and UPnP DM Control Point. CWMP requests received by the CWMP client from the ACS are translated by the TR-069/UPnP DM Proxy Module to the UPnP DM actions, and then passed to the UPnP DM Control Point to be sent to the UPnP DM devices. When an UPnP action response or event is received by the UPnP DM Control Point, the action response and event is passed to the TR-069/UPnP DM Proxy Module to be converted to a CWMP response or sent to the ACS using the CWMP event notification mechanism.

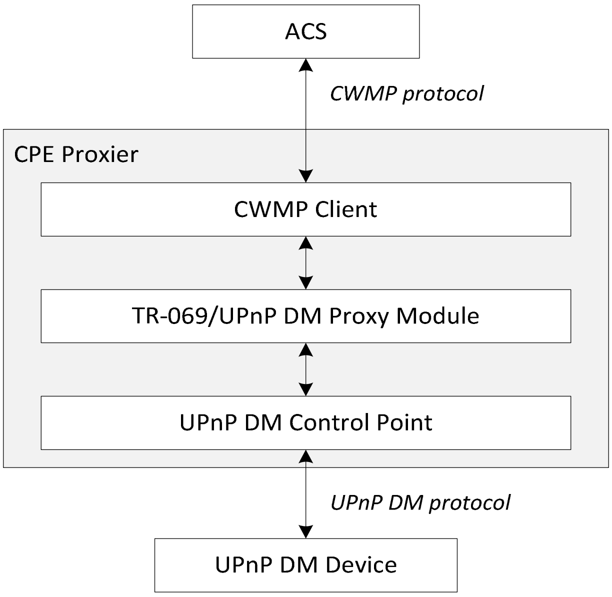


Figure 5.2.8: TR-069/UPnP DM Proxy Management Architecture

5.2.4.3.2 ZigBee® Proxy

Figure 5.2.9 and figure 5.2.10 present the principle and an example basic sequence for the management of ZigBee® devices by using TR-069 with the ZigBee® data model.

The ZigBee® devices reside behind a GW and communicate with the ACS via this GW. The GW resides normally in a CPE such as a broadband router (home gateway or business gateway). The GW has a proxy function to change a CWMP message to a ZDO function invocation based on the ZigBee® data model object. The proxy function changes messages by referring to a mapping of ZigBee® data model objects and CWMP methods to ZDO functions and their parameters. A management example is shown in figure 5.2.10.

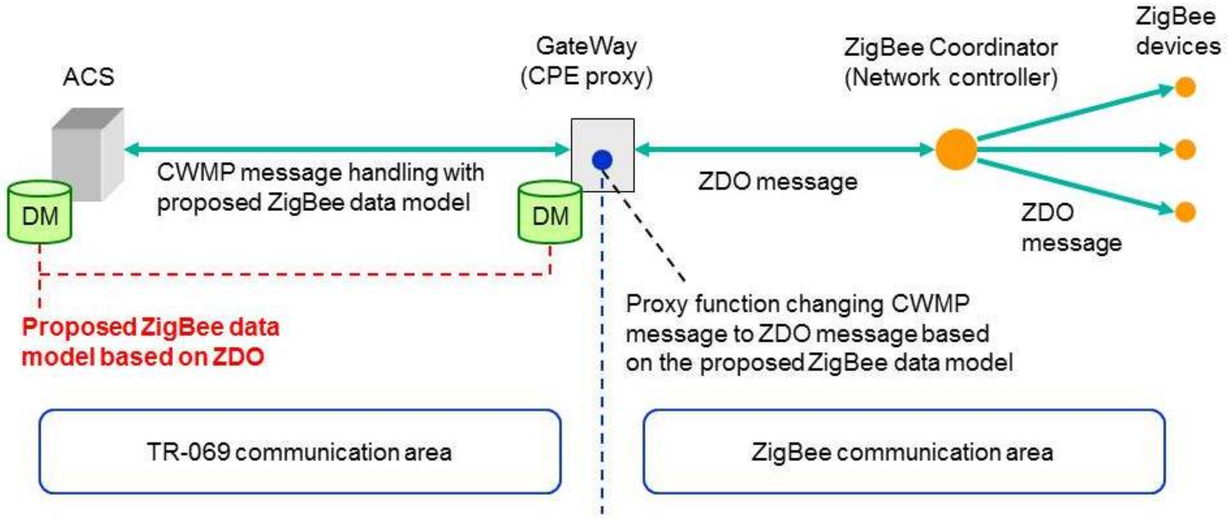

ACS
GateWay
(CPE proxy)
ZigBee Coordinator
(Network controller)
ZigBee
devices
CWMP message handling with
proposed ZigBee data model
DM
DM
ZDO message
ZDO
message
Proposed ZigBee data
model based on ZDO
Proxy function changing CWMP
message to ZDO message based
on the proposed ZigBee data model
TR-069 communication area
ZigBee communication area

Figure 5.2.9: Usage of the data model to manage ZigBee® devices with TR-069 [i.13]

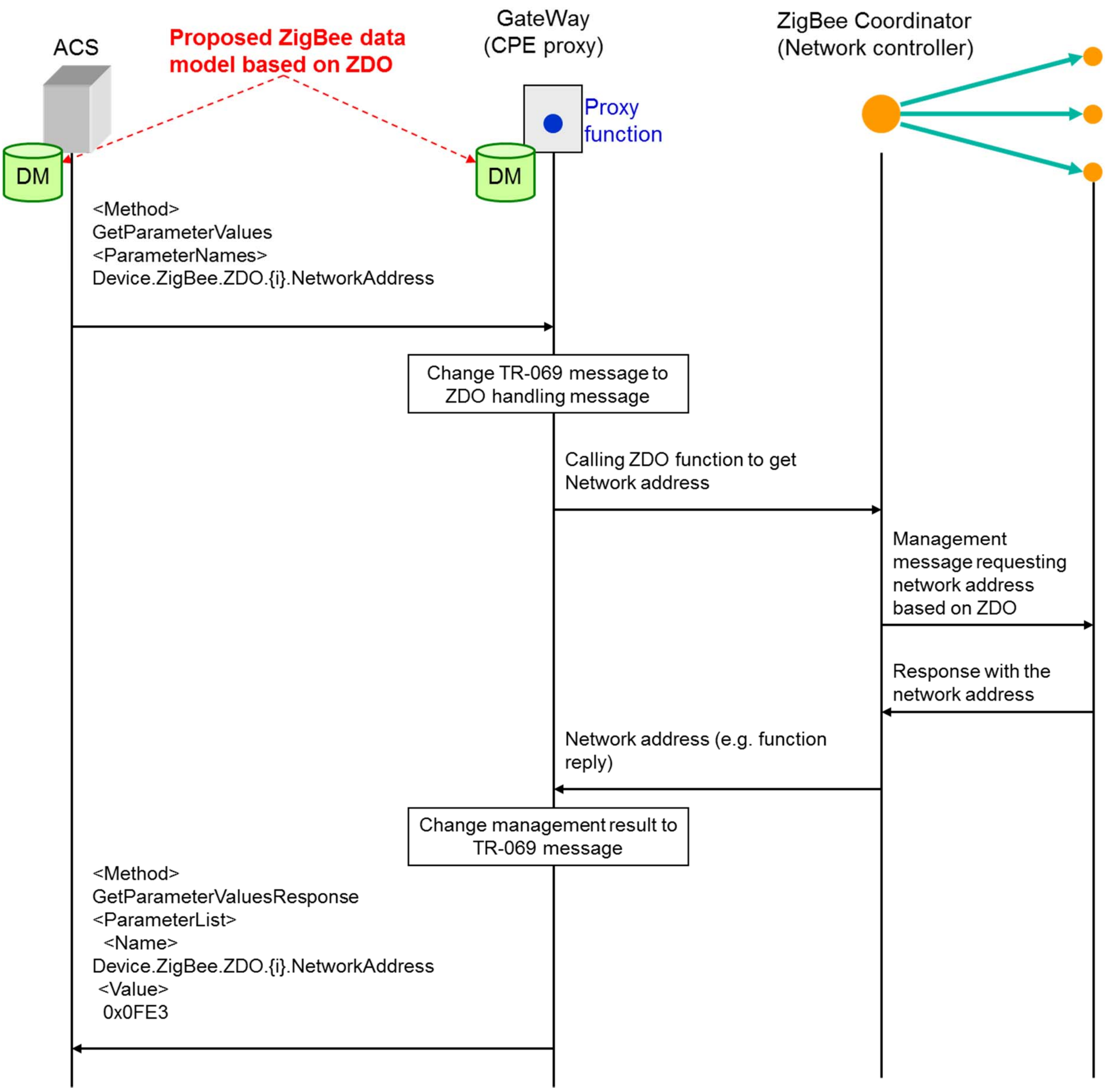

ACS
Proposed ZigBee data model based on ZDO
GateWay (CPE proxy)
ZigBee Coordinator (Network controller)
Proxy function
DM
DM
<Method> GetParameterValues <ParameterNames> Device.ZigBee.ZDO.{i}.NetworkAddress
Change TR-069 message to ZDO handling message
Calling ZDO function to get Network address
Management message requesting network address based on ZDO
Response with the network address
Network address (e.g. function reply)
Change management result to TR-069 message
<Method> GetParameterValuesResponse <ParameterList> <Name> Device.ZigBee.ZDO.{i}.NetworkAddress <Value> 0x0FE3

Figure 5.2.10: Example sequence diagram of ZigBee® management with TR-069

This example shows how the ACS gets a ZigBee® device's network address by using TR-069 [i.13] communication based on the ZigBee® data model. The ACS sends a CWMP message which includes the "GetParameterValues" as a method and the part of the ZigBee® data model "Device.ZigBee®.ZDO.{i}.NetworkAddress", which refers to the network address, as a parameter name. The proxy function in the GW changes the received message to a ZDO handling message to call some ZDO function on the ZC. The ZC manages the ZigBee® devices according to the called ZDO function and sends the result (the searched network address, in this case) to the proxy. The proxy function changes the ZDO management result to a CWMP message which is denoted in figure 5.2.10 as "GetParameterValuesResponse". The name of the parameter list is "Device.ZigBee®.ZDO.{i}.NetworkAddress" and the value of the parameter list is "0x0fE3" (network address instance).

5.2.5 Functions

The TR-069 family of specifications is intended to support a variety of functionalities to manage a collection of devices, including the following primary capabilities:

- Auto-configuration and dynamic service provisioning.
- Software/firmware image management.
- Software module management.

- Status and performance monitoring.
- Diagnostics.
- Proxy Management.
- Bulk data collection.

5.3 Introduction to OMA LightweightM2M (LWM2M)

5.3.1 Description

OMA Lightweight M2M is a protocol for device and service management for M2M. The main purpose of this technology is to address service and management needs for constrained M2M devices, over UDP and SMS bearers. The crucial aspects in this work are the:

- Target devices for this protocol are resource constraint devices (e.g. 8-16bit MCU, RAM is in tens of KB and flash is in hundreds of KB).
- Ability to perform Data collection and remote control of devices without the need for complex computing and UI operations.
- Optimization of network resources to allow a large numbers of devices may be connected to the communication network simultaneously.
- Fusion of device functionalities management and service manipulation into a single protocol.

From the implementation view LWM2M has the following features:

- Suitable for resource constraint devices.
- Usage of compact binary packets.
- Support for multiple data encoding formats that include Binary, JSON, plain text and opaque data formats.
- Easy to be implemented though the reuse of existing implementation of IETF technologies (e.g. CoAP).

One of typical use cases of using LWM2M technology is the firmware upgrade of streetlights [i.17].

1) A Streetlights supervisor is responsible for managing the streetlights system. (There are thousands of streetlights in the city and low-cost LWM2M devices embedded in the streetlights.)

2) The supervisor needs to remotely upgrade of the firmware of a specific streetlight or a group of streetlights.

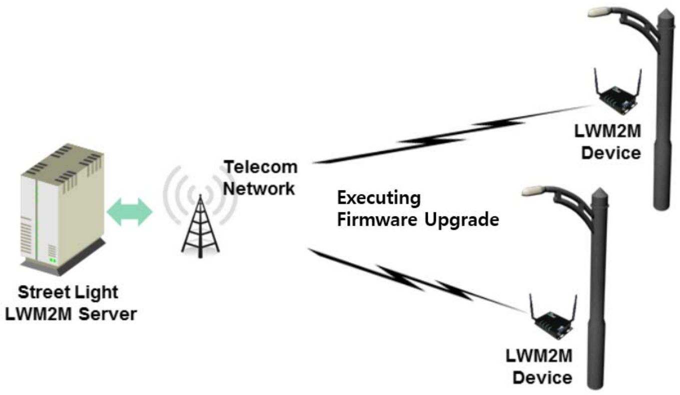


Figure 5.3.1: Firmware Upgrade of Streetlight of Use Case using LWM2M

5.3.2 Architecture

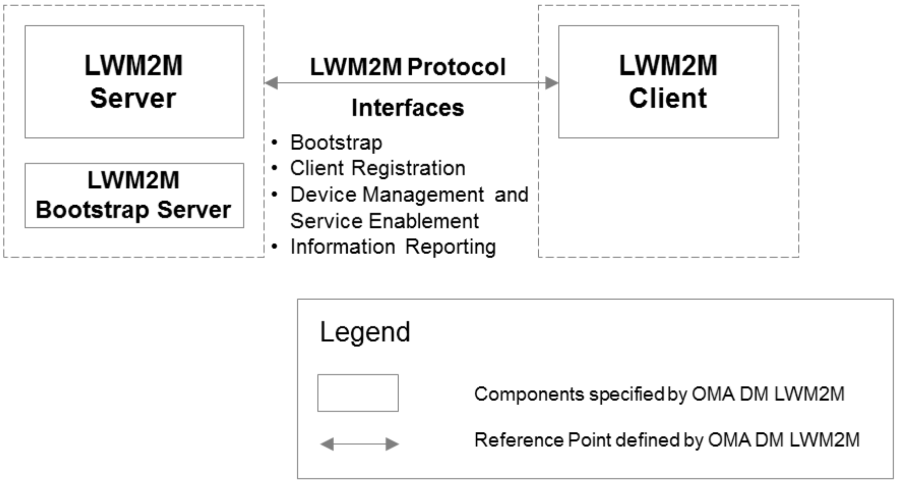


Figure 5.3.2: LWM2M Architecture

As shown in the figure 5.3.2, the layout is the architecture of LWM2M [i.18]. The Components specified by OMA LWM2M compose the LWM2M enabler which specifies the LWM2M Server/LWM2M Client interface. The LWM2M Server and LWM2M Client are typically instantiated in a M2M Server and a M2M Device.

Based on the deployment scenario, the LWM2M Server has the bootstrapping capability itself, or the LWM2M Bootstrap Server exists separately for security reasons.

5.3.3 Reference Points

This clause introduces the interfaces carried over the reference point consisting of two main components LWM2M Server and the LWM2M Client.

5.3.3.1 Functional Components

5.3.3.1.1 LWM2M Server

The LWM2M Server is a logical component which serves as an endpoint of the LWM2M protocol.

5.3.3.1.2 LWM2M Client

The LWM2M Client is a logical component. This LWM2M Client serves as an endpoint of the LWM2M protocol and communicates with the LWM2M Server to execute the device management and service enablement operations from the LWM2M Server and reporting results of the operations.

5.3.3.2 Interfaces

There are four interfaces supported by the reference point between LWM2M server and LWM2M Client. The logical operation of each interface is defined as follows:

- Bootstrap
  - This interface is used to provision essential information into the LWM2M Client so that the LWM2M Client can register to the LWM2M Server(s) after bootstrap procedure has completed.
- Client Registration
  - This interface allows the LWM2M Client register to the LWM2M Server. This procedure lets the Server know the existence and information (e.g. address, capabilities) of the LWM2M Client so that LWM2M Server can perform M2M services and device management on the LWM2M Client.

- Device Management and Service Enablement
    - This interface allows the LWM2M Server to perform the device management and M2M service enablement operations. Over this interface, the LWM2M Server can send operations to the LWM2M Client and gets response of the operations from the LWM2M Client.
- Information Reporting
    - This interface allows the LWM2M Client to report resource information to the LWM2M Server. This Information Reporting can be triggered periodically or by events (e.g. resource information is changed and configured conditions are met).

5.3.4 Protocols

5.3.4.1 Protocol Stack

The LWM2M has the protocol stack defined as below.

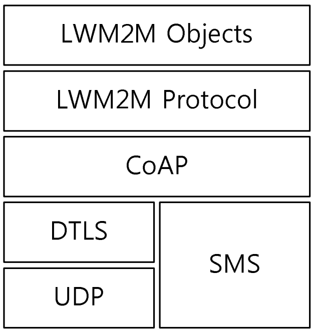


Figure 5.3.3: LWM2M Protocol Stack

- LWM2M Objects: LWM2M Objects are designed for the functionality provided by the LWM2M enabler. The LWM2M specification [i.19] defines a set of Standard Objects. Other Objects may also be added by OMA, external SDOs (e.g. the IPSO alliance) or vendors to enable certain M2M Services.
- LWM2M Protocol: LWM2M protocol defines the logical operations and mechanisms per each interface.
- CoAP: The LWM2M utilizes the IETF Constrained Application Protocol [i.20] as an underlying transfer protocol across UDP and SMS bearers. This protocol defines the message header, request/response codes, message options and retransmission mechanisms. The LWM2M only uses the subset of features defined in CoAP.
- DTLS: DTLS [i.21] is used to provide secure UDP channel between the LWM2M Server and the LWM2M Client for all the messages interchanged.
- UDP Binding with CoAP (Mandatory): Reliability over the UDP transport is provided by the built-in retransmission mechanisms of CoAP.
- SMS Binding with CoAP (Optional): CoAP is used over SMS by placing a CoAP message in the SMS payload using 8-bit encoding.

5.3.4.2 Resource Model

In the LWM2M Enabler technical specification [i.19], a simple resource model is described. Basically, a resource made available by resource model of the LWM2M Client is a Resource, and Resources are logically organized into Objects.

Figure 5.3.4 illustrates this structure, and the relationship between Resources, Objects, and the LWM2M Client. The LWM2M Client may have any number of Resources, each of which belongs to an Object.

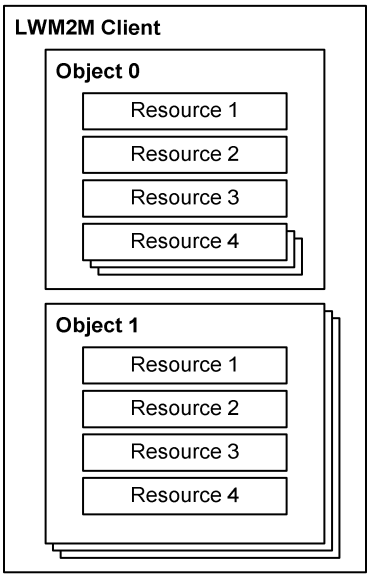


Figure 5.3.4: LWM2M Resource Model [i.19]

Resources are defined per Object, and each resource is given a unique identifier within that Object. Each Resource is designed to have one or more Operations that it supports. A Resource may contain multiple instances dependent on the Resource definition in Object specification.

An Object defines a grouping of Resources, for example the Firmware Object contains all the Resources used for firmware update purposes. The LWM2M enabler defines standard Objects and Resources and other Objects may be added to enable a certain M2M Services.

Object is instantiated either by the LWM2M Server or the LWM2M Client, which is called Object Instance before using the functionality of an Object. After Object Instance is created, the LWM2M Server can access that Object Instance and Resources in the Object Instance.

5.3.4.3 Interface Descriptions

5.3.4.3.1 Bootstrap

The Bootstrap interface is used to provision essential information into the LWM2M Client in order to allow the LWM2M Client to be able to register to a certain LWM2M Server. There are four modes for bootstrapping:

- Factory Bootstrap: the LWM2M Client is already provisioned at the time of the device manufacture. The pre-configured data is stored in the LWM2M Client.
- Bootstrap from Smartcard: When the Device supports a Smartcard and retrieval of bootstrap message from Smartcard is successful, the LWM2M Client processes the bootstrap message from the Smartcard and applies it to the LWM2M Client.
- Client initiated Bootstrap: the LWM2M Client retrieves the bootstrap message from a LWM2M Bootstrap Server. In this case the LWM2M Client needs to be pre-provisioned with the LWM2M Bootstrap Information before bootstrapping.
- Server initiated Bootstrap: the LWM2M Server provisions the bootstrap message into the LWM2M Client after recognizing the existence of the LWM2M Device. In this case the LWM2M Client needs to be pre-provisioned with the LWM2M Bootstrap Information before bootstrapping.

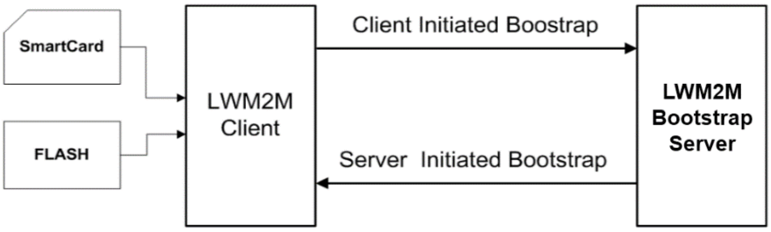


Figure 5.3.5: Bootstrap Modes

5.3.4.3.2 Client Registration

The Client Registration interface is used by the LWM2M Client to register with one or more LWM2M Servers, maintain each registration, and de-register from the LWM2M Server(s). When registering, the LWM2M Client indicates its Endpoint Name, MSISDN, supporting binding modes, lifetime of registration, the list of Objects the Client supports and available Object Instances. The registration is a soft state, with a lifetime indicated by the registration lifetime. The LWM2M Client periodically performs an update of its registration information to the registered Server(s). If the lifetime of a registration expires without receiving an update from the Client, the Server removes the registration information. Finally, when shutting down or discontinuing use of a Server, the Client performs de-registration.

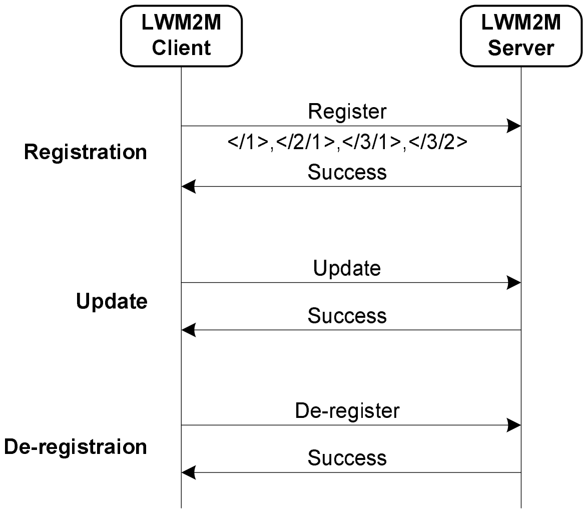


Figure 5.3.6: Example of Registration Procedure

5.3.4.3.3 Device Management and Service Enablement

This interface is used by the LWM2M Server to access Resources available from a LWM2M Client using Create, Read, Write, Delete, or Execute operations. The operations that a Resource supports are defined in the definition of its Object.

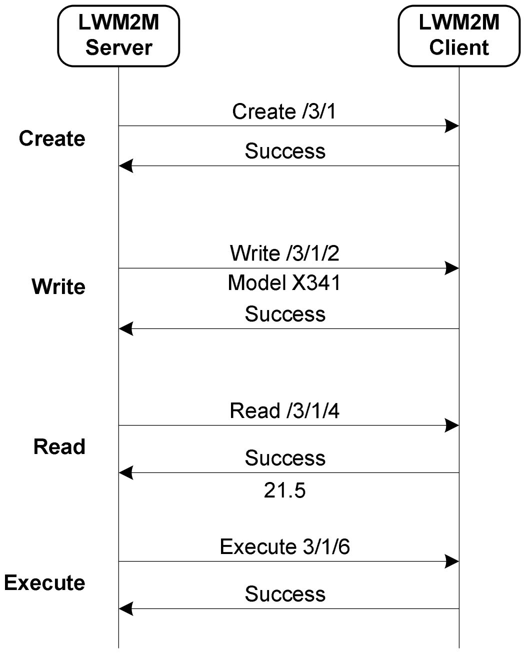


Figure 5.3.7: Example of Device Management and Service Enablement Interface

5.3.4.3.4 Information Reporting

This interface is used by the LWM2M Server to observe any changes in a Resource on the LWM2M Client, receiving notifications when new values are available. The LWM2M Server needs to configure observation related parameters by sending "Write Attribute" operation before observing Resources in the LWM2M Client. This observation relationship is initiated by sending an "Observe" operation to the L2M2M Client for an Object Instance or Resource. An observation ends when a "Cancel Observation" operation is performed or "Write Attribute" with cancel parameter operation is performed.

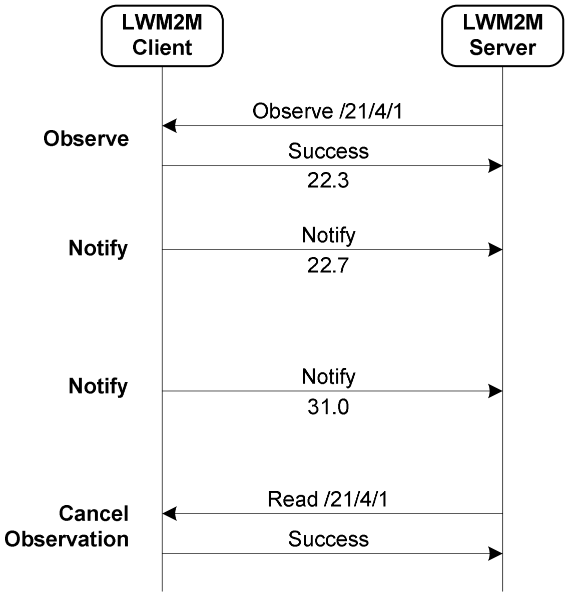


Figure 5.3.8: Example of Information Reporting Interface

5.3.5 Functions

A first set of standard Objects for the LWM2M 1.0 enabler have been developed. The Standard Objects are intended to support a variety of functionalities to manage LWM2M Devices. OMA may create further objects in future. Furthermore, other organizations and companies may define additional LWM2M Objects for their own M2M services using according to LWM2M Object Template and Guideline Annex in [i.19]:

- Server Security: security data related to the LWM2M server(s) and/or the LWM2M Bootstrap Server.
- Server: data, configuration, functions related to the LWM2M Server.
- Access Control: to check whether the LWM2M server has access right for performing an operation on Resources in the LWM2M Client.
- Device: provision of a range of device related information, device reboot and factory reset function.
- Connectivity Monitoring: to monitor parameters related to underlying network connectivity.
- Firmware: provision of firmware management, installing and updating new firmware.
- Location: provides location information of the LWM2M Devices.
- Connectivity Statistics: to statistical information of network connection (e.g. SMS counter, UDP data size).

5.4 Introduction to OMA Device Management 2.0

5.4.1 Description

OMA Device Management 2.0 is the new evolution of the OMA Device Management 1.x Protocols, and defines the interface between the DM Server and the DM Client to manage devices. Compared to OMA DM 1.x Protocols, the unique features of OMA DM 2.0 are as follows:

- Protocol Simplification and Optimization: Transaction model, DM Packages, addressing schemes and security model are simplified and optimized as well for the fast market penetrations. Simplifications and optimization is the fundamental spirits that goes throughout OMA DM 2.0.
- Extended DM Command Set: OMA DM 2.0 supports 10 DM commands, that is much extended compared to OMA DM 1.x. For example, the HGET/HPUT/HPOST commands are specified to utilize the RESTful interface, and the SHOW command is specified for the web-based user interaction.
- Management Data Delivery using HTTP (RESTful interface): The DM Client can retrieve or send the management data from or to the Data Repository using the HTTP protocol. This enables the effective management data delivery.
- Web-based User Interaction: The DM Server can utilize web pages to interact with the user. The Web Browser Component and the Web Server Component are newly introduced, and these two components can perform the user interaction session, which is independent with the DM session. This feature brings the rich user interactions to OMA DM 2.0.
- JSON Representation for DM Packages: In OMA DM 2.0, JSON replaces XML. The new JSON format for DM Packages lightens the parsing overheads and shortens the DM Package length keeping the same degree of the extensibility.
- New Addressing Scheme: OMA DM 2.0 introduces the new addressing scheme that uniquely addresses each node based on the Management Object Instance. In OMA DM 2.0, Management Objects do not need to be organized as a tree, and the DM Server does not need to have the knowledge of the DM Tree that can be different for each type of devices.

OMA DM 2.0 is not backward compatible with OMA DM 1.x. This is because OMA DM 2.0 uses the completely different representation based on JSON, and eliminates complex functionalities that are not required by the market. Although OMA DM 2.0 is not backward compatible with OMA DM 1.x, Management Objects (e.g. FUMO, SCOMO, DiagMon, LAWMO) designed for OMA DM 1.x can be still used for OMA DM 2.0. This is possible since OMA DM 2.0 uses the same Management Object definition and provides the necessary functionalities such as Generic Alerts. The complete separation between Management Objects and the OMA DM Protocol is one of the success factors as well.

5.4.2 Architecture

Figure 5.4.1 shows the OMA DM 2.0 Architecture and the related components and the interfaces.

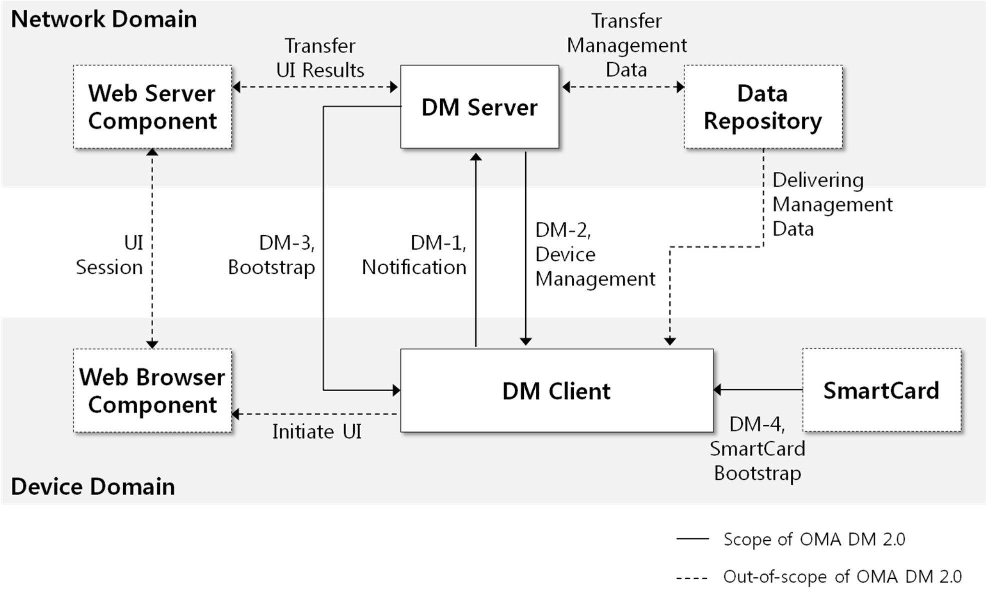


Figure 5.4.1: OMA DM 2.0 Architecture

The Web Server Component and the Web Browser Component can be utilized for the user interactions. For the user interaction, OMA DM 2.0 specifies that the DM Server can send the SHOW command; therefore, the DM Client initiates the Web Browser Component with the specified web page. The actual user interaction can be performed using the HTTP/HTML, which is out-of-scope of the OMA DM 2.0. Please note that how to initiate the Web Browser Component, and how the transfer the user interaction results are out-of-scope of OMA DM 2.0 as well.

Using the Data Repository, the DM Client can retrieve and upload the management data from or to the Data Repository using the HTTP protocol. The DM Server can send the HPUT/HPOST and HGET command for those purposes.

5.4.3 Reference Points

5.4.3.1 Functional Components

5.4.3.1.1 DM Client

The DM Client is the logical software component that conforms to the requirements for DM Clients specified in OMA DM 2.0.

5.4.3.1.2 DM Server

The DM Server is the logical software component that conforms to the requirements for DM Servers specified in OMA DM 2.0. The DM Server is also responsible for providing the Bootstrap Message using the DM-3 Interface.

5.4.3.1.3 Web Server Component

Web Server Component is the logical software component responsible to host web pages for the Web Browser Component in the device. The web pages are used for the user interactions.

5.4.3.1.4 Web Browser Component

Web Browser Component is the logical software component responsible for retrieving web pages from the Web Server Component and presenting them to the user.

5.4.3.1.5 Data Repository

Data Repository is the logical software component that the DM Client can retrieve or send management data to or from this component by using HTTP.

5.4.3.2 Interfaces

The brief explanations for the defined interfaces are as follows:

- DM-1 Notification: the interface over which the DM Server sends DM Notification to the DM Client to initiate the DM session.
- DM-2 Device Management: the interface over which the DM Server sends device management commands to the DM Client and the DM Client returns status code, results and Alerts.
- DM-3/4 Bootstrap: the interface over the Bootstrap Message is delivered. OMA DM 2.0 supports factory bootstrap, client-initiated bootstrap, server-initiated bootstrap and SmartCard bootstrap.

5.4.4 Protocol

5.4.4.1 DM Packages

Figure 5.4.2 describes the DM Package exchanges between the DM Client and the DM Server defined in OMA DM 2.0.

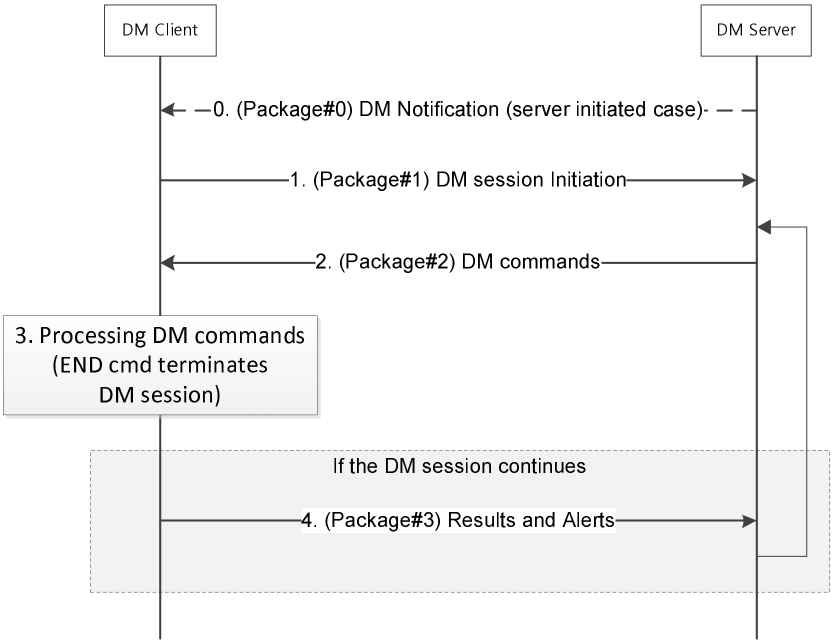

DM Client
DM Server
0. (Package#0) DM Notification (server initiated case)-
1. (Package#1) DM session Initiation
2. (Package#2) DM commands
3. Processing DM commands (END cmd terminates DM session)
If the DM session continues
4. (Package#3) Results and Alerts

Figure 5.4.2: OMA DM 2.0 Package Flow

- Step 0 (DM Notification): The DM Server requests the DM session by sending the DM notification to the DM Client.
- Step 1 (DM Session Initiation): This DM package initiates the DM session and might contain information for the supported Management Object and Generic Alerts generated by the DM Client.
- Step 2 (Request): The DM Server sends the DM commands to the DM Client.

- Step 3 (Command Processing): The DM Client processes the DM commands received. To process the DM command, the DM Client might interact with components such as the Web Browser Component, the Data Repository.
- Step 4 (Response): Unless the END command is received, the DM Client responses to the DM Server. It contains the results for the DM command and Generic Alerts generated by the DM Client.

5.4.4.2 DM Commands

Table 5.4.1 shows the DM commands specified in the OMA DM 2.0.

Table 5.4.1: OMA DM 2.0 Commands

Logical Op.	Name	Description	DM 1.x Relation
Read	GET	To retrieve data from the device. The data is included in Package#3.	Get
	HPUT	To request the device to send data to the Data Repository using HTTP PUT.	MO individually implements this
	HPOST	To request the device to send data to the Data Repository using HTTP POST.	
Write	DELETE	To delete data in the device.	Delete
	HGET	To requests the DM Client to retrieve data from the Data Repository using HTTP GET, and add or replace the received data into the device.	Add, Replace
Exec	EXEC	To execute an executable node.	Exec
Not related to MO	SHOW	To initiate a UI session between the Web Browser Component and the Web Server Component.	Alert for UI. Only 5 UI types exist
	CONT	To continue the DM session with the specified DM Server URI.	<RespURI> element
	END	To terminate the DM session.	<Final> element
	DEFAULT	To use a specific address to capture configuration if that is missing in the device for a specific MOID.	None

5.4.5 Functions

5.4.5.1 Introduction

Each device that deploys OMA DM 2.0 supports Management Objects. Management Object is the set of related nodes for a specific function, and OMA DM 2.0 achieves management functions by using the Management Objects. The type of the Management Object is defined by the MOID.

For example, for the firmware management functions, the Firmware Update Management Object as specified in [i.4] can be used as shown below.

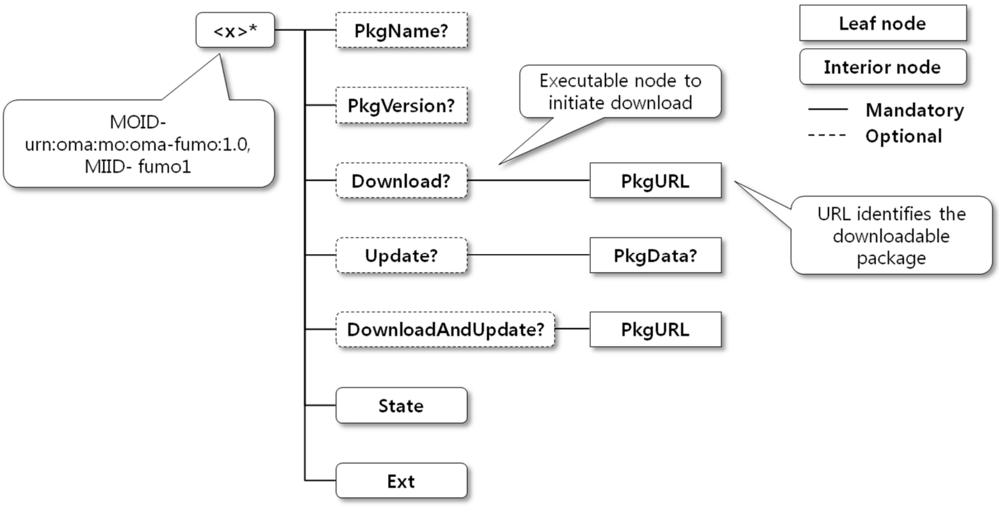


Figure 5.4.3: Firmware Update Management Object

OMA DM 2.0 does not require that Management Objects in the device are organized as a hierarchical tree structure since the nodes are addressed based on the Management Object instance. In the device, Management Object is realized as a Management Object instance. Once the Management Object instance is created in the device, the DM Client assigns the MO Instance Identifier (MIID) to it, and the DM Server can use the ClientURI to uniquely identify each node in the Management Object Instance. Note that ClientURI is built based on the MOID and MIID.

Suppose that the above FUMO is realized as an instance in the device. Then the DM Server can use below Client URIs:

- urn:oma:mo:oma-fumo:1.0/fumo1/PkgName - to identify the <x>/PkgName node.
- urn:oma:mo:oma-fumo:1.0/fumo1/DownloadAndUpdate/PkgURL - to identify the <x>/DownloadAndUpdate/PkgURL node.

The "fumo1" in the above ClientURI is the MIID assigned to the FUMO instance, and can be omitted if the MOID is enough to uniquely identify a Management Object Instance (i.e. there is only one Management Object Instance for the MOID). Hence, the ClientURI can be simplified further to "urn:oma:mo:oma-fumo:1.0/ /PkgName" if MIID is not needed.

5.4.5.2 Management Objects supported by OMA DM 2.0

OMA DM 2.0 shares the same Management Object definitions, which allows that Management Objects designed for OMA DM 1.x can be reused for OMA DM 2.0. Hence, every MO identified in the clause 5.1.5.2.2 that is considered as relevant to the management of M2M Device/Gateway are supported by the OMA DM 2.0 (e.g. FUMO, SCOMO, Gateway MO, etc.). In addition, OMA DM 2.0 supports below standard MOs.

Table 5.4.2: Standard Management Object for OMA DM 2.0

Standard MO	Description
DevInfo MO V1.2	This MO provides the basic device information such as the device identifier, manufactures, etc.
DM Account MO V2.0	This MO provides the account information for the DM Servers.
Delegation Access Control MO V1.0	This MO provides the information for the access control.
Session Info MO V1.0	This MO provides the information for the DM session.

5.4.5.3 Detailed Comparisons with OMA DM 1.x

Table 5.4.3 explains the differences between OMA DM 2.0 and OMA DM 1.x from the overall perspective.

Table 5.4.3: Comparison between OMA DM 2.0 and OMA DM 1.x

Label	DM 2.0	DM 1.x
Package Representation	JSON	XML
DM Command	10 DM commands Sequential processing only	6 DM commands. Random, Sequential, Atomic processing
MO Addressing	ClientURI	Absolute, Relative, Virtual
User Interaction	Web-based User Interaction using SHOW command	User Interaction Alert supports 5 types (Display, Confirmation, User Input, Single Choice, Multi Choice)
Security	Authorization - ACL for MO instance granularity (can be extended to node granularity) Confidentiality/Authentication - transport layer security only	Authorization - ACL for node granularity Confidentiality/Authentication - DM Protocol security or transport layer security
DM Notification	TLV (2 Headers, 5 Options) Transport: SMS, Google Cloud Messaging	TLV (6 Headers, 8 Options, Digest) + Binary Format (backward compatible with DM 1.2) Transport: SMS, OBEX, SIP, HTTP, Cell Broadcast

5.4.5.4 Protocol Examples

In this clause, an example is presented in which the DM Server updates the firmware of the device. The Package#1 (DM session initialization) sent from the DM Client to the DM Server is as follows:

```
POST /dmclient/dm20 HTTP/1.1
Content-Type: application/vnd.oma.dm.initiation+json
Accept: application/vnd.oma.dm.request+json
OMADM-DevID: IMEI:493005100592800
Host: www.dms.com

{
    "MOS": [
        {
            "DDF": "http://www.vendor.com/DDF/devinfo1.0.ddf",
            "MOID": "urn:oma:mo:oma-dm-devinfo:1.0",
            "MIID": ["miid1"]
        },
        {
            "DDF": "http://www.vendor.com/DDF/oma-fumo1.0.ddf",
            "MOID": "urn:oma:mo:oma-fumo:1.0",
            "MIID": ["miid1"]
        },
        {
            "DDF": "http://www.vendor.com/DDF/oma-dm-dmacc2.0.ddf",
            "MOID": "urn:oma:mo:oma-dm-dmacc:2.0",
            "MIID": ["miid1"]
        }
    ]
}
```

In the above Package#1, the JSON object "MOS" carries the Management Object information that is supported by the device. According to the "MOS", the DM Server can know that the DM Client supports DevInfo MO, FUMO, DM Account MO.

After receiving the Package#1, the DM Server sends the Package#2 to update the firmware of the device. The Package#2 is as follows:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.oma.dm.request+json

{
    "CMD": [
        ["EXEC", "oma:mo:oma-fumo:1.0//Update"]
    ]
}
```

After completing the firmware updating, the DM Client sends the Package#3 for the response as follows:

```
POST /dmserver/dm20 HTTP/1.1
Content-Type: application/vnd.oma.dm.response+json
Accept: application/vnd.oma.dm.request+json
OMADM-DevID: IMEI:493005100592800
Host: www.dms.com

{
    "Status": [
        {"sc": 200}
    ]
}
```

After receiving the Packge#3 from the DM Client, the DM Server terminates the DM session by sending the END command. The Package#2 for this is as follows:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.oma.dm.request+json

{
    "CMD": [
        ["END"]
    ]
}
```

On receiving the Package#2, the DM Client does not send the Package#3. The DM session is terminated.

6 Gap analysis of existing relevant technolgoies

6.1 Management related requirements gap analysis reference

The definitions for the values in below table are:

- FULL: the requirement can be fulfilled by the technology alone.
- PARTIAL: the requirement can be partially fulfilled by the technology.
- ALLOWED: Adopting this technology will allow this requirement to be implemented.
- NOT SUPPORTED: This technology does not fulfil the requirement AND adopting this technology would not allow the requirement to be implemented.

Table 6.1.1: Requirements fulfilment reference

Requirement Support				
	OMA DM 1.3	**BBF TR-069**	**OMA LWM2M**	**OMA DM 2.0**
MGR-001	Partial	Partial	Full	Partial
MGR-002	Full	Full	Full	Full
MGR-003	Full	Full	Full	Full
MGR-004	Allowed	Allowed	Allowed	Allowed
MGR-005	Partial	Partial	Partial	Partial
MGR-006	Full	Full	Full	Full
MGR-007	Full	Full	Full	Full
MGR-008	Full	Full	Partial	Full
MGR-009	Full	Full	Full	Full
MGR-010	Partial	Partial	Partial	Partial
MGR-011	Full	Full	Full	Full
MGR-012	Full	Full	Full	Full
MGR-013	Allowed	Allowed	Allowed	Allowed
MGR-014	Full	Full	Full	Full
MGR-015	Full	Full	Full	Full
MGR-016	Full	Full	Full	Full
MGR-017	Not Supported	Not Supported	Not Supported	Not Supported

6.2 MGR-001

6.2.1 Requirement Description

The M2M System shall support management and configuration of M2M Gateways/Devices including resource constrained M2M Devices [i.22].

NOTE: See the annex A as a guidance about the definition of resource constrained and what kinds of the existing management technologies are suitable to apply the constrained devices.

6.2.2 OMA DM 1.3

OMA DM 1.3 provides PARTIAL support for this requirement.

OMA DM 1.3 requires an OMA DM compliance device shall have at least one of the protocol stacks among TCP/IP, IrDA or WSP. And the devices shall also have a capability to parse the xml file. Because the DM Representation OMA DM uses to deliver the DM Message is in the format of XML. The OMA DM devices shall also be capable of store a certain amount of information which is the MO trees to carry the management functions. For constrained devices that serve very simple functions and have the basic capability of parsing short XML and small amount of storage to store the MO, OMA DM 1.3 can be used for device management. As a result, OMA DM can be applied to some resource constrained devices but not those very limited in resources (no memory, cannot parse the XML, no communication module).

OMA DM 1.3 can also configure devices with DM Client using the ClientAPI between DM Client and the local application. With the local application defined by oneM2M, devices can be configured using service layer protocols.

6.2.3 BBF TR-069

TR-069 [i.13] provides PARTIAL support for this requirement.

The TR-069 [i.13] provides support for resource constrained devices that are CWMP enabled through the use of its standard CWMP protocols. For resource constrained devices that are not CWMP enabled (e.g. ZigBee® devices, IP devices without CWMP stack), TR-069 [i.13] provides mechanisms to access the constrained devices through a CWMP enabled device called a CWMP Proxy. Clause 5.2.2.1 TR-069 Proxy Management describes this architecture. A technology constraint exists in that the CWMP Proxy must have connectivity, typically LAN, with the non-CWMP enabled device. As such, the TR-69 [i.13] Proxy Management functions generally reside on a M2M Gateway within the customer premises.

Resource constrained devices that are CWMP enabled requires, at a minimum, the support for the:

- Protocol stack as defined in clause 5.1.4.1 ACS to CPE Protocol.
- Implementation of the TR-181i2 Baseline:3 profile [i.29].

Resources required to implement a CWMP stack have been advertised as low as 150 Kilobytes storage and 30 Kilobytes DRAM (heap and stack) on an Android operating system.

Many resource constrained devices require monitoring of the device's environment (e.g. processor, memory, battery, temperature), the TR-181i2 data model provides support for many of these objects (processor, memory, temperature) where these objects may be monitored and alarmed using the FaultMgmt objects of the data model or using the Active/Passive notification mechanism described in section 3.7.1.5 of TR-069 [i.13]. While TR-181i2 provides support for many objects within a resource constrained device, the current data model does not provide support for a Battery resource. This type of resource may be implemented using Vendor specific extensions or submitted to the Broadband Forum for inclusion in a revision of the TR-181i2 data model.

6.2.4 OMA LWM2M

OMA LWM2M provides FULL support for this requirement.

Since the main focusing M2M device of LWM2M is resource constraint device, LWM2M is specialized in managing and configuring resource constraint devices.

For resource constraint device, LWM2M has several features which are listed below:

- CoAP with minimum set of required features for header and options.
- Either UDP or SMS as transport layer binding.
- Binary TLV format (Tag, Length, Value) for conveying values of multiple Resources.
- JSON format is optionally supported.

6.2.5 OMA DM 2.0

OMA DM 2.0 provides PARTIAL support for this requirement.

The device that conforms to OMA DM 2.0 shall support TCP/IP, HTTP protocol stack, which might not be applicable for severely resource constrained devices (e.g. 8-bit microcontrollers with small accounts of memory).

OMA DM 2.0 also requires the HTTP client in the device that can be used to retrieve management data from the Data Repository. Note that this is a mandatory feature of OMA DM 2.0. Optionally, OMA DM 2.0 might require the Web Browser Component in the device to support the web-based user interaction.

If the resource constrained device can support TCP/IP and HTTP protocol, OMA DM 2.0 can be used to manage those devices with the simple DM package representations based on the JSON format.

Compared to OMA DM 1.3, OMA DM 2.0 has different factors to support resource constrained devices as follows:

- Supporting only HTTP transport-binding.
- Providing the simple JSON-based DM package representations.
- Requiring the HTTP Client to interact with the Data Repository.

Optionally requiring the Web Browser Component for the user interaction.

6.3 MGR-002

6.3.1 Requirement Description

The M2M System shall provide the capability to discover the M2M Area Networks including information about devices on those networks and the parameters (e.g. topology, protocol) of those networks [i.22].

6.3.2 OMA DM 1.3 and OMA DM 2.0

OMA DM 1.3 provides FULL support for this requirement.

In the setup phase of the OMA DM protocols. The MO DevInfo is transported from DM Client to DM Server. And the MO DevDetail can be requested by DM Server if necessary. In the MO DevInfo and DevDetail, information about how the device can be reached, the protocol, the address, port number, required security parameters is transferred from device to DM server.

For devices in the local area network that are attached to the DM Gateway, GwMO can be used to get the address information.

Also some work has been done in ETSI M2M to define MANMO and MANDMO [i.23] to enable the DM Server to get the information about the topology and protocol of the local area network.

In this way, the DM Server can get to know the connection related parameters (protocol) from the device.GwMO defines how DM Server can manage device in a local area network through DM Gateway. DM Gateway can work in three modes which are transparent mode, proxy mode and adaptor mode. OMA DM devices and non-OMA DM devices can all be managed using DM Gateway. Combined with other MOs defined by OMA DM, devices in the local area network can be managed by DM Server.

6.3.3 BBF TR-069

TR-069 [i.13] provides FULL support for this requirement.

TR-069 [i.13] provides support for discovery of devices in the associated Local Area Networks for CWMP enabled devices.

TR-069 [i.13] proxy management has mechanisms where a CPE Proxier discovers devices using device discovery mechanisms as described in Appendix I of TR-069 [i.13]. These mechanisms rely on the discovery of the device using the device's native protocol (e.g. UPnP DM, ZigBee®, Z-wave).

The discovery of the topology of the area network in which the device exists is constrained by the device's native protocol support for topology discovery. For example - UPnP DM does not provide any support for discovery of topologies while ZigBee® topologies can be inferred by evaluating the routing tables of the ZigBee® nodes. The TR-181 [i.29] data model exposes these elements (e.g. ZigBee® routing tables) but rely on the management systems to develop the topologies.

The TR-181i2 data model [reference TR-181 [i.29] Device Data Model for TR-069 Issue: 02 Amendment 6 November 2012 provides support for the following LAN topologies:

- Ethernet.
- WiFi.
- USB.
- HPNA.
- MoCA.
- G.hn.
- HomePlug.
- Universal Powerline Association (UPA).
- UPnP.

In addition the ZigBee® Pro topology support is expected to be included in the next release of TR-181i2.

6.3.4 OMA LWM2M

OMA LWM2M provides FULL support for this requirement.

Connectivity Monitoring Object contains which networks are available and which network is currently used.

Also, it has parent router IP address of each M2M Device so the M2M System can discover entire topology of M2M Local Area Network.

Protocol (e.g. WLAN, Bluetooth®) used by M2M Device is specified in Connectivity Object also.

6.4 MGR-003

6.4.1 Requirement Description

The M2M System shall provide the capability to maintain and describe the management information model of devices and parameters (e.g. topology, protocol) of M2M Area Networks [i.22].

6.4.2 OMA DM 1.3 and OMA DM 2.0

OMA DM 1.3 provides FULL support for this requirement.

For devices in the local area network, some work has been done in ETSI M2M to define MANMO and MANDMO [i.23] to maintain and describe the information model about the topology and protocol of the local area network. The MANMO and MANDMO have been submitted for registration in OMA.

6.4.3 BBF TR-069

TR-069 [i.13] provides FULL support for this requirement.

TR-069 [i.13] provides capabilities to describe and maintain the management information model of CWMP and non-CWMP enabled devices as through the Supported Data Model and Software/Firmware Management features of the protocol. Within TR-069 [i.13] all devices and services that are of interest to the problem space are modelled using the TR-069 [i.13] XML meta-model. These models are a description of the device and services that are under management.

These models can be either configured within the device or CWMP Proxy (if the device is not CWMP enabled) or the device can report its Supported Data Model to the ACS.

6.4.4 OMA LWM2M

OMA LWM2M provides FULL support for this requirement.

Connectivity Monitoring Object contains lots of Resources to maintain and describe the information model about the topology and protocol of the local area network.

LWM2M has several features to be able to manage LWM2M Clients in M2M Local Area Network. The main barrier is how to reach LWM2M Clients from LWM2M Server:

- Let a LWM2M Server know when IP Address is changed at LWM2M Client by sending Registration Update logical operation.
- Support Queue mode which makes LWM2M Server queue the request until LWM2M Client is online. If LWM2M Client with Queue mode is within M2M Local Area Network, LWM2M Client sends Update message triggered by time or event. After LWM2M Server receives Update message, the LWM2M Server reaches LWM2M Client so the LWM2M Server sends queued message.

6.5 MGR-004

6.5.1 Requirement Description

The M2M System shall support common means to manage devices enabled by different management technologies (e.g. OMA DM, BBF TR-069) [i.22].

6.5.2 OMA DM 1.3 and OMA DM 2.0

OMA DM 1.3 can ALLOW the fulfilment of the requirement.

OMA DM does not provide features that translate between OMA DM and other management protocols (e.g. BBF TR-069, oneM2M). As such there is an expectation that the M2M System would translate between management protocols.

The management of devices defined by OMA DM is fulfilled by sending DM Messages from DM Server to DM Client. The management related information is carried by Management Objects. For the oneM2M system to support common means to manage devices through OMA DM, the oneM2M system could include the DM Server, DM Client and DM Gateway in the oneM2M system and could have some abstraction to include the MO trees in the oneM2M system to enable the device management in spite of the detailed technologies.

6.5.3 BBF TR-069

TR-069 [i.13] ALLOWS fulfilment of the requirement by Service Layer mechanisms.

TR-069 [i.13] family of specifications do not provide features that translate between TR-069 and other management protocols (e.g. OMA-DM, oneM2M). As such there is an expectation that the M2M System would translate between management protocols.

TR-069 [i.13] provides access to manage devices through its Auto-configuration Server. The Auto-configuration Server has interfaces with the NMS/OSS and BSS systems of the Service provider. As such the ACS would have expected to interface with the M2M System.

6.5.4 OMA LWM2M

OMA LWM2M ALLOWS fulfilment of the requirement when common means between the oneM2M Server and LWM2M Server are defined. OMA LWM2M only defines the protocol between the LWM2M Server and LWM2M Client.

LWM2M does not provide features that translate between LWM2M and other management protocols (e.g. OMA-DM, oneM2M, TR-069).

6.6 MGR-005

6.6.1 Requirement Description

The M2M System shall provide the capability to manage multiple devices in a grouped manner [i.22].

6.6.2 OMA DM 1.3 and OMA DM 2.0

OMA DM 1.3 provides PARTIAL support for this requirement.

The requirement can be fulfilled both in service layer and by OMA DM technology.

For service layer, one DM Server could manage multiple DM Clients. The oneM2M system is able to manage multiple devices in a grouped manner by utilize one DM Server. The M2M System could have multiple devices in a group. When the M2M System needs to send identical DM Message to each device in the group, the M2M System could pass the command to DM Server to send DM Messages one by one or simultaneously.

For manage group of devices in OMA DM. GwMO is defined how to fan out DM Command to a group devices.

6.6.3 BBF TR-069

TR-069 provides PARTIAL support for this requirement.

TR-069 family of specifications defines management actions that are destined for a single CWMP enabled device. The concept of groups within the TR-069 family of specifications does not exist. Grouping, while not specified within the CWMP protocol is implemented within ACS or the NMS/OSS/BSS systems. As such, the M2M System would be required to implement the grouping feature.

6.6.4 OMA LWM2M

OMA LWM2M provides PARTIAL support for this requirement.

The requirement can be fulfilled at service layer.

A LWM2M Server could manage multiple LWM2M Clients. oneM2M system is able to manage multiple devices in a grouped manner by utilizing one LWM2M Server. The M2M System could have multiple devices in a group. When oneM2M System needs to send identical message to each device in the group, the oneM2M System could pass the command to the LWM2M Server to send the messages one by one.

6.7 MGR-006

6.7.1 Requirement Description

The M2M System shall provide the capability for provisioning and configuration of devices in M2M Area Networks [i.22].

6.7.2 OMA DM 1.3 and OMA DM 2.0

OMA DM 1.3 provides FULL support for this requirement.

For devices in the local area network, GwMO can be used to provision and configure End Devices that are attached to the DM Gateway. Gateway Config MO contains the information related to each attached End Device.

6.7.3 BBF TR-069

TR-069 provides FULL support for this requirement.

TR-069 provides support for provisioning and configuration of devices in the Local Area Networks for CWMP enabled and non-CWMP enabled devices. TR-069 has the capability to manage a device through the device's life-cycle (bootstrap through decommissioning).

For CWMP enabled devices that reside behind a Gateway with Firewalls and Network Address Translation features enabled, the CWMP protocol provides a mechanism that allows the ACS to communicate those devices. This mechanism is described in as described in annex G of TR-069 [i.13].

6.7.4 OMA LWM2M

OMA LWM2M provides FULL support for this requirement.

In LWM2M, bootstrap mechanism is used to provision and configure the M2M Device. Four approaches are introduced in TS: Manufacturer Pre-configuration, SmartCard Provisioning, Client Initiated Bootstrap, and Server Initiated Bootstrap. And detail bootstrap steps are described.

6.8 MGR-007

6.8.1 Requirement Description

The M2M System shall provide the capability for monitoring and diagnostics of M2M Gateways/Devices in M2M Area Networks [i.22].

6.8.2 OMA DM 1.3 and OMA DM 2.0

OMA DM 1.3 provides FULL support for this requirement.

The capability of monitoring and diagnostics of OMA DM is mainly achieved by MO DiagMon. DiagMon supports diagnostics policies management, fault reporting, performance monitoring, device interrogation, remote diagnostics procedure invocation and remote device repairing. The monitoring and diagnostics of the devices in the local area network can be fulfilled by DiagMon plus GwMO.

6.8.3 BBF TR-069

TR-069 provides FULL support for this requirement.

TR-069 provides support for monitoring and diagnostics of devices in the Local Area Networks for CWMP enabled and non-CWMP enabled devices. Monitoring can include notification support for devices that need immediate attention as well as passive monitoring of device information and statistics that may be collected in a periodic manner. Likewise, TR-069 provides the capability to execute diagnostics in a synchronous or asynchronous fashion; allowing for long lived diagnostics to be executed on a device,

For non-CWMP enabled devices, the device is implemented as a non-CWMP enabled Virtual Device where the procedure is documented in appendix I of TR-069 [i.13].

CWMP and non-CWMP enabled Virtual and Embedded devices support the following diagnostic operations:

- Reboot.
- Factory Reset.

In addition CWMP enabled and non-CWMP enabled Virtual Devices also support the following standard diagnostics:

- IP Diagnostics (Ping, Trace Route, HTTP or FTP Download or Upload, UDP Echo).
- DNS - NS Lookup.
- HPNA Diagnostics.
- UPA Diagnostics.
- Device Self Tests.
- DSL Line.
- ATM Interface.

These tests are documented within the TR-181i2 data model [i.29].

6.8.4 OMA LWM2M

OMA LWM2M provides FULL support for this requirement.

In LWM2M, Connectivity Monitoring Object and Connectivity Statistics Object are used to monitor current connection and collect connection information to configure parameters based on the collected result respectively.

LWM2M enabler also has concept to send diagnostic information such as GPS module failure, IP connectivity failure, and peripheral malfunction.

6.9 MGR-008

6.9.1 Requirement Description

The M2M System shall provide the capability for software management of devices in M2M Area Networks [i.22].

6.9.2 OMA DM 1.3 and OMA DM 2.0

OMA DM 1.3 provides FULL support for this requirement.

Software management capability is fulfilled by the MO SCOMO (Software Component Management Object). SCOMO can be used to remotely manage a software component within a device. The functionalities provided by SCOMO includes delivery, download, installation, update, removal, activation, and de-activation of software. The software management of devices in local area network can be fulfilled by SCOMO plus GwMO.

6.9.3 BBF TR-069

TR-069 provides FULL support for this requirement.

TR-069 provides support for software and firmware management of devices in the Local Area Networks for CWMP enabled and non-CWMP enabled devices. Software and firmware management of non-CWMP enabled devices may be performed using Software modules within the CWMP enabled M2M Gateway or may be downloaded directly to device by implementing the non-CWMP enabled device as a Non-CWMP enabled Virtual Device.

6.9.4 OMA LWM2M

OMA LWM2M provides PARTIAL support for this requirement.

Since LWM2M talks about resource constraint M2M devices, LWM2M enabler provides the capability for management of a full package, firmware. So firmware contains all the software which is necessary for all the services provided by M2M Service Platform.

6.10 MGR-009

6.10.1 Requirement Description

The M2M System shall provide the capability for rebooting and/or resetting of M2M Gateways/Devices and other devices in M2M Area Networks [i.22].

6.10.2 OMA DM 1.3 and OMA DM 2.0

OMA DM 1.3 provides FULL support for this requirement.

The functionality for reset the device is provided by the MO LAWMO. LAWMO is for remotely locking and wiping the device. The functionality of rebooting a device is enabled by DiagMon. In this way, OMA DM can reset the device to its original state and reboot the device as well. The reset and reboot of a device in local area network can be enabled by DCMO or DiagMon plus GwMO.

6.10.3 BBF TR-069

TR-069 provides FULL support for this requirement.

TR-069 provides support for resetting (boot, factory) of devices in the Local Area Networks for CWMP enabled and non-CWMP enabled devices. For non-CWMP enabled devices, the device is implemented as a Non-CWMP enabled Virtual Device.

In the scenario where the device in the M2M Local Area network is a CWMP enabled device, TR-069 provides a mechanism where the device can communicate through a Gateway (which might be NAT or Firewall enabled) to the ACS. Annex G of TR-069 [i.13] describes this mechanism.

6.10.4 OMA LWM2M

OMA LWM2M provides FULL support for this requirement.

Device Object has a Reset Resource which is used for resetting M2M Devices. LWM2M also provides factory reset function which makes M2M Device initial state of deployment.

6.11 MGR-010

6.11.1 Requirement Description

The M2M System shall provide the capability for authorizing devices to access M2M Area Networks [i.22].

6.11.2 OMA DM 1.3 and OMA DM 2.0

OMA DM 1.3 provides PARTIAL support for this requirement.

Configuration methods of OMA DM can be used to detach certain device from the network. Coordinator and router can also be configured to block the new access request to deny devices to be attached to the local area network. In this way, devices can be authorized to access M2M Local Area Network.

6.11.3 BBF TR-069

TR-069 provides PARTIAL support for this requirement.

TR-069 does not provide direct support for authorizing devices for access to Local Area Networks but does allow for configuration of credentials and other properties that these networks utilize. TR-069 does this for CWMP and non-CWMP enabled devices.

6.11.4 OMA LWM2M

OMA LWM2M provides PARTIAL support for this requirement.

If M2M devices have configuration for accessing M2M Local Area Networks prior to deployment, the M2M devices can be authorized based on the configured information.

6.12 MGR-011

6.12.1 Requirement Description

The M2M System shall provide the capability for modifying the topology of devices in M2M Area Networks [i.22].

6.12.2 OMA DM 1.3 and OMA DM 2.0

OMA DM 1.3 provides FULL support for this requirement.

OMA DM can be used to modify the topology of devices in M2M Local Area Network by activate and de-activate devices that serve as coordinator or router in the area network. For example, if a coordinator or router is de-activated, devices attached to the coordinator or router will automatically find other coordinators or routers to access the local area network.

Configuration methods can also be used to configure the router or coordinator to accept or deny access request of new devices. The topology is modified.

6.12.3 BBF TR-069

TR-069 provides FULL support for this requirement.

TR-069 does allow for configuration of properties for Local Area Networks that can determine which devices are to be included within a Local Area Network. TR-069 does this for CWMP and non-CWMP enabled devices.

6.12.4 OMA LWM2M

OMA LWM2M provides FULL support for this requirement.

OMA LWM2M can be used to modify the topology of devices in M2M Local Area Network by disabling M2M devices which serve as coordinator in the area network. For example, if a coordinator is disabled, devices in M2M Local Area Network attached to the coordinator will automatically find other coordinators to access the local area network.

6.13 MGR-012

6.13.1 Requirement Description

Upon detection of a new device the M2M Gateway shall be able to be provisioned by the M2M Service Infrastructure with an appropriate configuration which is required to handle the detected device [i.22].

6.13.2 OMA DM 1.3 and OMA DM 2.0

OMA DM 1.3 provides FULL support for this requirement.

In a DM Server assisted bootstrap procedure defined in GwMO. Whenever a new device is detected, DM Server will install the End Device credentials to the gateway through Gateway Config MO.

In OMA DM 1.3, DM Gateways with DM Client can be also provisioned using the ClientAPI between DM Client and the local application. With the local application defined by oneM2M, devices can be configured using service layer protocols.

6.13.3 BBF TR-069

TR-069 provides FULL support for this requirement.

TR-069 provides support for detection of new devices in the Local Area Networks. TR-069 has the capability to detect new devices via active and passive notification mechanisms described in section 3.7.1.5 of TR-069 [i.13] as well as using the CWMP protocol's inform event (e.g. Bootstrap) mechanisms for CWMP enabled devices.

In the most common M2M deployment scenarios, the M2M Gateway would include a CWMP Agent making the M2M Gateway a CWMP-enabled device with a CWMP Proxier as defined in annex J of TR-069 [i.13].

In this scenario, the M2M Gateway's CWMP agent could detect and report the new device to the ACS via the M2M Gateway's CWMP agent. Once a device is reported to the ACS, the ACS can inform the M2M System about the device addition for further configuration and software/firmware management activities. Possible configuration activities could include:

- Software management (annex H of TR-069 [i.13]).
- Device configuration via Proxy management (appendix I of TR-069 [i.13]).

6.13.4 OMA LWM2M

OMA LWM2M provides FULL support for this requirement.

LWM2M does not have gateway concept, so to be able to handle the detected devices, the M2M Gateway needs to have LWM2M Server feature which means that device thinks gateway as LWM2M Server. In this case, M2M Gateway can manage the device detected.

6.14 MGR-013

6.14.1 Requirement Description

The M2M System shall be able to identify and manage M2M Service status of M2M Devices [i.22].

6.14.2 OMA DM 1.3 and OMA DM 2.0

OMA DM 1.3 and 2.0 ALLOWS fulfilment of the requirement by development of a oneM2M Service Layer data model for management purposes.

DCMO defined in OMA DM can be used to enable or disable device capabilities, such as hardware, IO and connectivity. OMA DM 1.3 and 2.0, as device management protocols, are able to manage the M2M Service layer using the DM protocol to configuration and retrieval of M2M Services that execute within M2M Devices. The only thing necessary is that a data model of the M2M Service be defined for management purposes.

6.14.3 BBF TR-069

TR-069 ALLOWS fulfilment of the requirement by development of a oneM2M Service Layer data model for management purposes.

The TR-069 family of specifications, as a device management protocol, is able to manage the M2M Service layer using the CWMP protocol to configuration and retrieval of M2M Services that execute within M2M Devices. The only thing necessary is that a data model of the M2M Service be defined for management purposes.

6.14.4 OMA LWM2M

OMA LWM2M ALLOWS fulfilment of requirement by development of a M2M Service Layer data model for management purposes.

The LWM2M specification, as a device management protocol, is able to manage the M2M Service layer using the LWM2M protocol to configuration and retrieval of M2M Services that execute within M2M Devices. The only thing necessary is that a data model of the M2M Service be defined for management purposes.

6.15 MGR-014

6.15.1 Requirement Description

The M2M System shall be able to retrieve events and information logged by M2M Gateways/ Devices and other devices in M2M Area Networks [i.22].

6.15.2 OMA DM 1.3 and OMA DM 2.0

OMA DM 1.3 and 2.0 provides FULL support for this requirement.

Related technologies are defined in DiagMon:Trap which describes how the DM Client monitors the performance of the device. The DiagMon Client can send notification to DM Server or collect trap event together to response the retrieve request. As a result, OMA DM 1.3 and 2.0 can fully fulfill the requirement.

With regard to devices in the M2M Area Network, DiagMon is combined with GwMO to fulfill the requirement.

6.15.3 BBF TR-069

TR-069 provides FULL support for this requirement.

The TR-069 family of specifications, as a device management protocol, is able to retrieve events and logs for M2M Gateway/Devices and devices in M2M Area Networks through the use of the CWMP protocol using the standard data model and proxy mechanisms. Information can be retrieved using the CWMP get RPC as well as file transfer mechanisms (e.g. FTP, HTTP). In addition, the IPDR protocol can be used to retrieve information that would be considered bulk transfer.

6.15.4 OMA LWM2M

OMA LWM2M provides FULL support for this requirement.

The LWM2M Client has capability to collect error or event information such as GPS module failure, out of memory, SMS failure, and low battery power. The LWM2M has functionality to collect data to be notified due to subscription when the LWM2M Client is offline or the LWM2M Server is temporarily disabled.

6.16 MGR-015

6.16.1 Requirement Description

The M2M System shall be able to support firmware management (e.g. update) of M2M Gateways/ Devices and other devices in M2M Area Networks [i.22].

6.16.2 OMA DM 1.3 and OMA DM 2.0

OMA DM 1.3 and 2.0 provides FULL support for this requirement.

It is defined in FUMO about how to support firmware management. FUMO defined in OMA DM can be used to initiate firmware update, exchange device information, download update package, install update package, notify firmware update.

With regard to devices in the M2M Area Network, FUMO is combined with GwMO to fulfill the requirement.

6.16.3 BBF TR-069

TR-069 provides FULL support for this requirement.

The TR-069 family of specifications, as a device management protocol, is able to download firmware for M2M Gateway/Devices and devices in M2M Area Networks through the use of the CWMP protocol using the standard data model and proxy mechanisms. Firmware can be retrieved using the CWMP download RPC file transfer mechanisms (e.g. FTP, HTTP). One constraint exists in the proxy mechanism for downloading firmware to device in a M2M Area Network. A device in a M2M Area Network must be a Virtual Device to have allow for the download RPC to be utilized.

6.16.4 OMA LWM2M

OMA LWM2M provides FULL support for this requirement.

LWM2M has Firmware Object to update firmware of the LWM2M Client. LWM2M supports two download mechanisms: directly write firmware package, and give URI and let the LWM2M Client download firmware package. After updating firmware, LWM2M has capability to inform the connected LWM2M Servers of what functionalities are added in the LWM2M Client.

6.17 MGR-016

6.17.1 Requirement Description

The M2M System shall be able to retrieve information related to the Static and Dynamic Device/Gateway Context for M2M Gateways/Devices as well as Device Context for other devices in M2M Area Networks [i.22].

6.17.2 OMA DM 1.3 and OMA DM 2.0

OMA DM 1.3 and 2.0 provides FULL support for this requirement.

DiagMon defined by OMA DM 1.3 and 2.0 enable the DM Server to acquire information related to the devices local context including battery level, available memory as well as some network capabilities.

DiagMon also defines Trap method which can be used to collect events related to the change of dynamic context in the device. The collected events can be retrieved by the DM Server.

ClientAPI defined in OMA DM 1.3 and 2.0 also provides interfaces that can be used to retrieve MO information from DM Client which can be static device context.

6.17.3 BBF TR-069

TR-069 provides FULL support for this requirement.

The TR-069 family of specifications, as a device management protocol, is able to retrieve device specific information for M2M Gateway/Devices and devices in M2M Area Networks through the use of the CWMP protocol using the standard data model and proxy mechanisms. As the TR-069 model utilizes data models for representing information that is easily extended either through standardization activities or vendor specific attributes.

6.17.4 OMA LWM2M

OMA LWM2M provides FULL support for this requirement.

LWM2M has Static Context such as manufacturer, model number, and serial number, and Dynamic Context such as battery level, memory free, location, time, IP address, current network (e.g. WCDMA, GSM, LTE, Bluetooth®, WiFi®), and serving cell id. LWM2M has capability to expose parent IP address, which can make topology of M2M Area Network.

6.18 MGR-017

6.18.1 Requirement Description

The M2M system shall support the capability to map M2M service subscription role(s) to roles used within technology specific Device Management protocols [i.22].

6.18.2 OMA DM 1.3 and OMA DM 2.0

This requirement is NOT SUPPORTED by OMA DM 1.3 and 2.0.

OMA DM does provide security capabilities (ACLs) [i.33] within the DM Server to ensure authorized parties are permitted access control of a Management Object. However, OMA DM uses server identifier to distinguish different DM servers which cannot be mapped to any concept of roles. Since role is not supported by OMA DM, the requirement is not supported by OMA DM.

6.18.3 BBF TR-069

This requirement is NOT SUPPORTED by the TR-069 family of specifications.

The TR-069 family of specifications does provide security capabilities (ACLs) within the CPE to ensure authorized parties are permitted access control of an Object. However this ACL mechanism does not discern different management roles within the ACS. The ACL mechanism in the CPE treats the ACS as the one authorized party. In addition, a CPE can only report to one ACS at a time. As such, the TR-069 ACS mechanism does not provide security authorization of resources to various roles within the ACS as expectation is, since a CPE can only be managed by one ACS, that this function is the responsibility of the ACS.

Likewise the ACL capability, while specified, is not widely supported in CPEs. In fact the BBF certification program does not include ACLs in its current certifications suite of tests.

Also the TR-069 family of specifications does not specify an interface northbound of the ACS except that TR-131 [i.28] does provide guidance to northbound systems such as M2M systems. If an interface with associated Service Layer to Device Management Layer security mechanisms are required, the expectation would be that oneM2M would either specify an interface between the ACS or work collaboratively with the Broadband Forum to specify a northbound interface from an ACS for the purposes of M2M Service Layer interaction.

6.18.4 OMA LWM2M

This requirement is NOT SUPPORTED by OMA LWM2M.

OMA LWM2M provides an access control mechanism based on an ACL in order for the LWM2M Client to authorize operations on a certain Resource/Object sent from the LWM2M Server. However, OMA LWM2M does not have any concept of roles; the ACL contains identification of the LWM2M Server only. Since role concept is not supported by OMA LWM2M, the requirement is not supported by OMA LWM2M.

7 Device Management Deployment Scenarios

7.1 Introduction

This clause describes the deployment scenarios currently utilized in deployments and proposed future deployments of the Device Management technologies listed in the present document.

7.2 Current Management Deployment Scenarios

This clause describes the common deployment scenarios that exist today for deployments of the Device Management technologies listed in the present document.

7.2.1 Managed Device Using Network Operator Management

When discussing M2M Device Management deployment scenarios we must review the Device Management deployment scenarios that exist today in the underlying network operator's communication network. In the scenario depicted below the underlying network operator has, other than the end user, exclusive control of the resources within the device. Also in this deployment scenario, the device management technologies described in the present document utilize a management client in the device that connects to the underlying network operator's management server. Typically there is one instance of the management client within the device that connects to one management server controlled by the underlying network operator. In several of the device management technologies the management client in the device offers a proxy capabability that provides the underlying network operator with the capability to manage devices that do not have their own management client. The underlying network operator's Operational Support System(s) (OSS) manage devices through their interaction with the underlying network operator's management server.

The underlying network operator ensures exclusive control of the resources within the device by providing device firmware and software that have been approved for use by the underlying network operator.

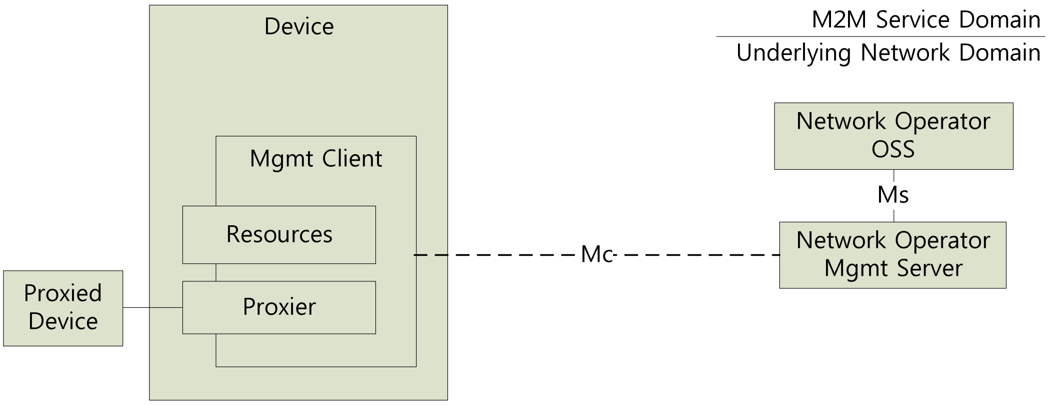


Figure 7.2.1: Device Management in the Communication Network

7.2.2 Managed Device Using Service Provider Management

With the introduction of the M2M System a new stakeholder, the M2M Service Provider, also requires control of resources of the device. In this scenario depicted below, the M2M Service Provider controls all or selected resources of the device via its own management server which may be part of the M2M Service Platform. In this scenario the underlying network operator ensures that the M2M Service Provider has control of underlying network operator restricted resources using an out-of-band responsibility delegation mechanism. The delegation mechanism utilized is usually a certification process by the underlying network operator of the firmware and software that is downloaded on the device by the M2M Service Provider. The process of certifying the software and firmware of the device is outside the scope of the present document.

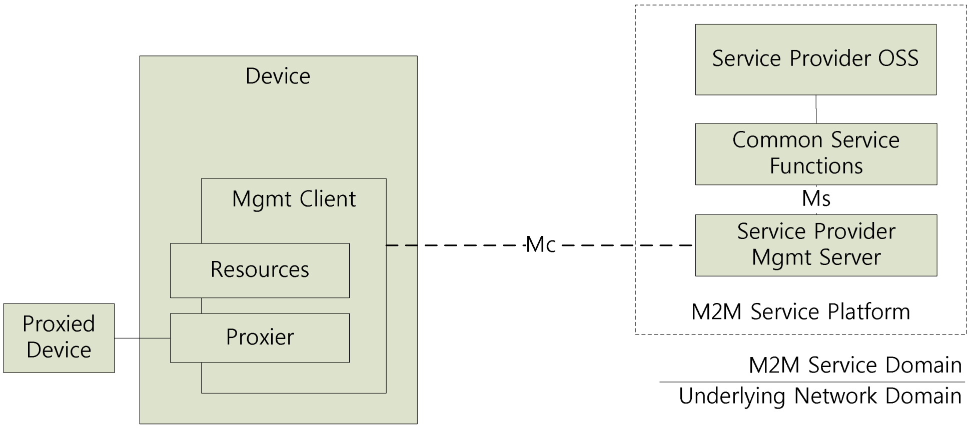


Figure 7.2.2: Service Provider Controlled Devices

7.3 Possible Future Management Deployment Scenarios

This clause describes common deployment scenarios that are expected to exist in the future for deployments of the Device Management technologies listed in the present document in addition to the ones described above.

7.3.1 Shared Managed Device Using Network Operator Management

In some scenarios, the underlying network operator and the M2M Service Provider share control of the device by utilizing the underlying network operator's management server. The underlying network operator restricts which resources the M2M Service Provider can control by exposing the capabilities from the management server to the M2M Service Platform. Typically this is performed by providing the M2M Service Provider an account, with appropriate access control to the resources, within the underlying network operators management server.

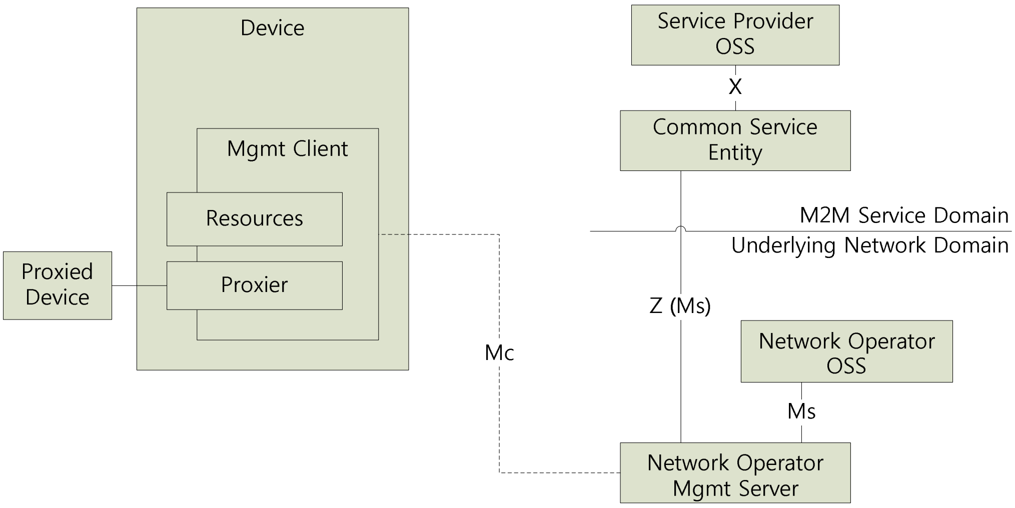


Figure 7.3.1: Shared Control via Network Operator Management Server

7.3.2 Shared Managed Device Using Service Provider Management

In some scenarios, the underlying network operator and the M2M Service Provider share control of the device by utilizing the service provider's management server. The service provider's management server is the primary DM server, and any requests that the underlying network operator wants to initiate on the M2M device using occurs over the X interface to the service provider's Common Services Entity, which in turn sends the request to the service provider's DM server who will execute the DM operation. This scenario is typically interesting when the M2M Service Provider uses different underlying networks that all require some "unified" DM access to the M2M devices (example: the use case "Oil and Gas Pipeline Cellular/Satellite Gateway from TR-0001 Use Cases).

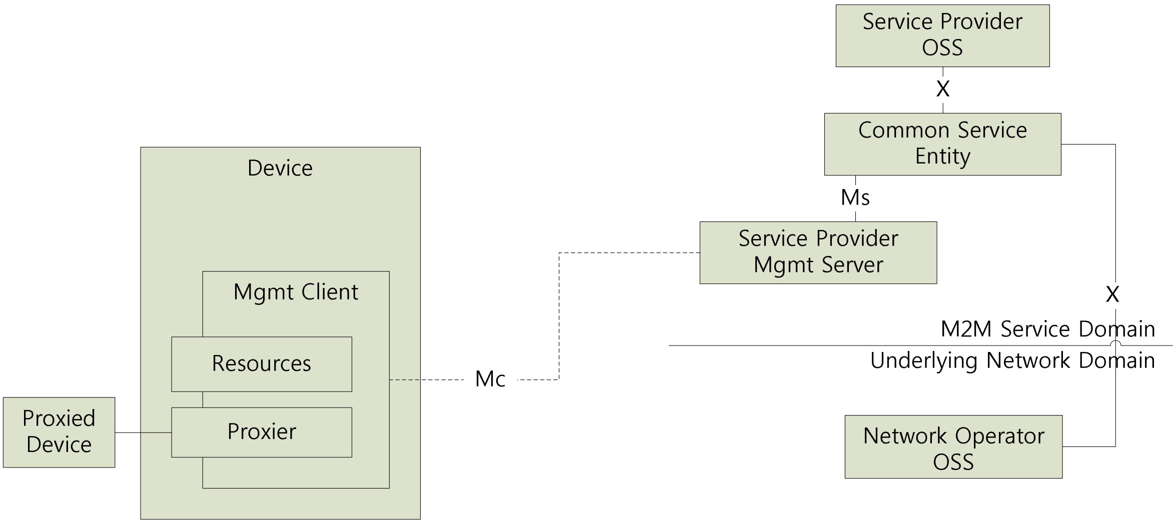


Figure 7.3.2: Shared Control via Service Provider Management Server

7.3.3 Shared Managed Device Using Separate Management

In this scenario the underlying network operator and the M2M Service Provider share control of the device by utilizing separate management servers as depicted by the below figure. It should be noted that some technologies (e.g. TR-069) cannot implement the scenario as a management client can only connect to one management server. In this scenario, the authorization enforcement point can be implemented within the device using the delegation and access control list features of the device management technologies or in the M2M Service Platform. Regardless of the enforcement point, the network operator restricts control to the network operator controlled resources by certifying the firmware and software that is placed on the device.

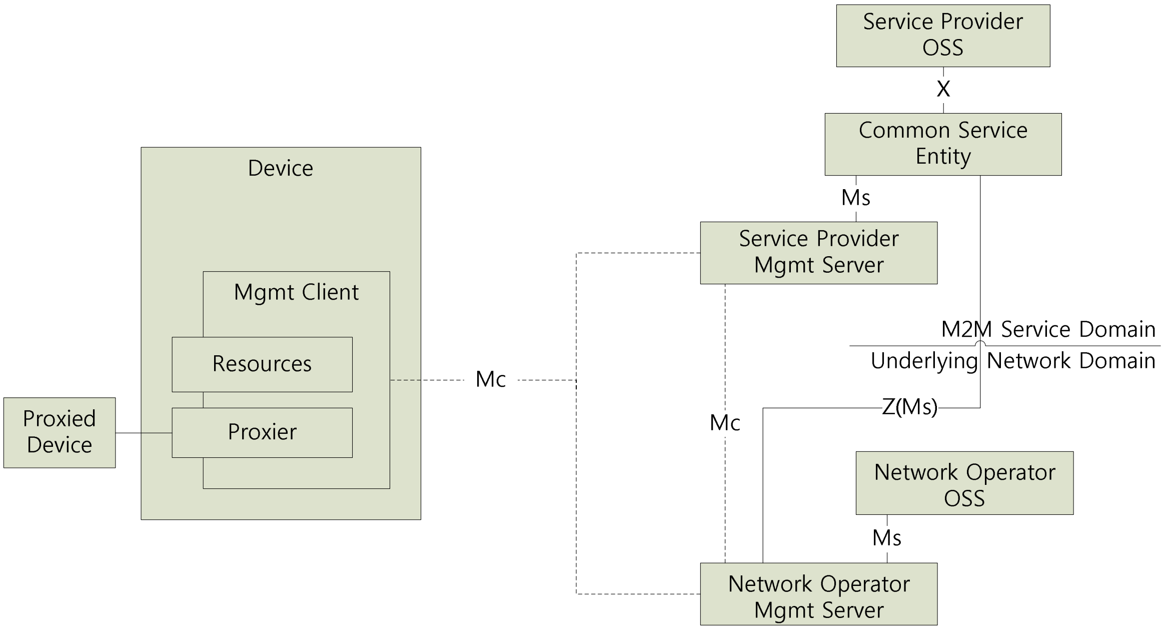


Figure 7.3.3: Shared Control via Separate Management Servers

7.3.4 Federated Managed Device Using Separate Management

In some M2M Devices, the control of resources in a device is federated within separate operating environments of the device as depicted by the figure below. This scenario is typical of devices with multiple CPU complexes where one complex is dedicated to the access network termination functions (e.g. machine termination) while another CPU complex is dedicated to application functions. One important point of this type of deployment scenario is fact that there are multiple management clients where a management client is assigned to the M2M Service Provider's management server and another management client is assigned to the underlying network operator's management server.

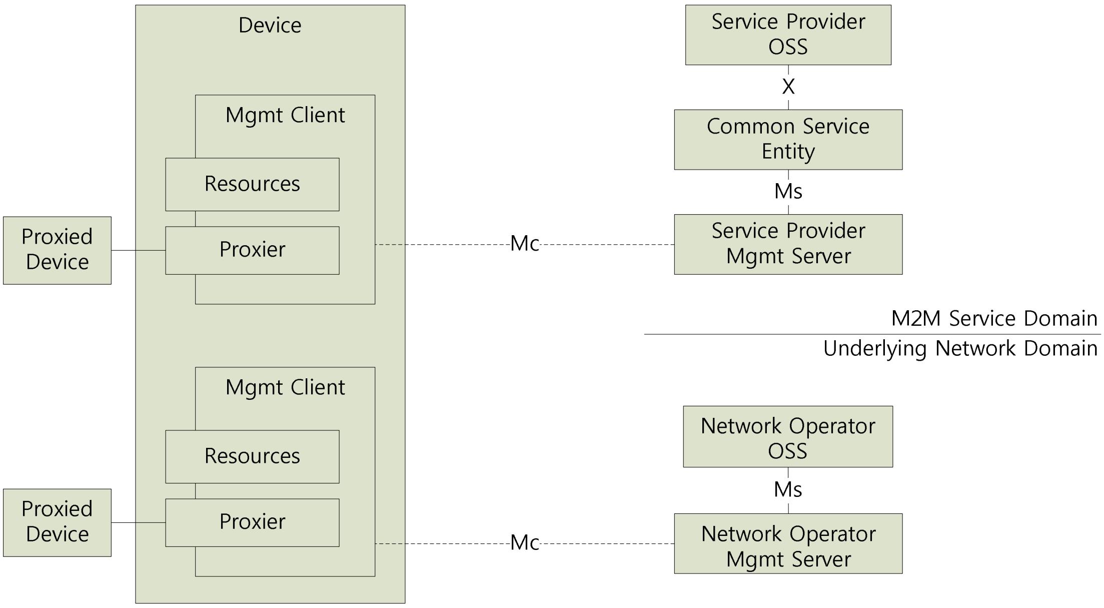


Figure 7.3.4: Federated Managed Device

7.3.5 Conclusions To Guide the Device Management Architecture

The deployment scenarios described in the clause 7.2 allows several conclusions can be described to guide the development of the device management architecture in M2M Systems. These conclusions are:

The M2M Service Provider manages resources in a device by:

- Utilizing the Network Operator's management server to access resources in the device.
- Operating a separately owned management server that allows access to resources in the device. In this case, a separate management client might exist in the device to which the M2M Service Provider's management server connects.
- The M2M Service Platform reaches the Network Operator or M2M Service Provider management server through the Z reference point.

Resources within a device are owned by either the Network Operator or the M2M Service Provider. Access to the resource is controlled by:

- Certifying the firmware or software that is used on the device
- Utilizing the access control mechanisms of the device management technologies (e.g. accounts, delegation, access control lists).

7.4 Architectural Framework Considerations

The consideration of the device management architectural framework takes into account different deployment scenarios and has been leveraged into the Functional Architecture technical specification [i.34]. More details can be found in the description of device management CSF within the Functional Architecture technical specification.

Annex A: Guidance for Managing Resource Constrained Devices

The oneM2M specification is expected to support devices that are resource constrained. As seen clearly in the clause 6 of the present document, different types of constrained devices, according to memory and processing capabilities, can be managed by using different management technology. In this annex, the proper definition what resource constrained devices are and what kinds of management technologies are suitable to apply the constrained devices are discussed as a guideline.

A.1 Classification of Resource Constrained Devices

One of controversial issues is that there is no clear definition of 'resource constrained devices' and 'lightweight or heavy devices'. However, there is a useful reference [i.25] to the definition of constrained devices. It can be used for enabling the classification of devices according to the RAM and storage usage for implementing protocol stacks to apply the management technologies since OMA DM 1.3/2.0, OMA LWM2M, and BBF TR-069 [i.13] utilize the different binding protocols stacks.

[i.25] defines some succinct terminology for different classes of constrained devices. The table below represents the criteria of each classes based on the memory constraints.

Table A.1.1

Name	Data Size (e.g. RAM)	Code Size (e.g. Flash)
Class 0	<< 10 kilobytes	<< 100 kilobytes
Class 1	~ 10 Kilobytes	~ 100 kilobytes
Class 2	~ 50 kilobytes	~ 250 kilobytes
Class 3	>> 50 kilobytes	>> 250 kilobytes

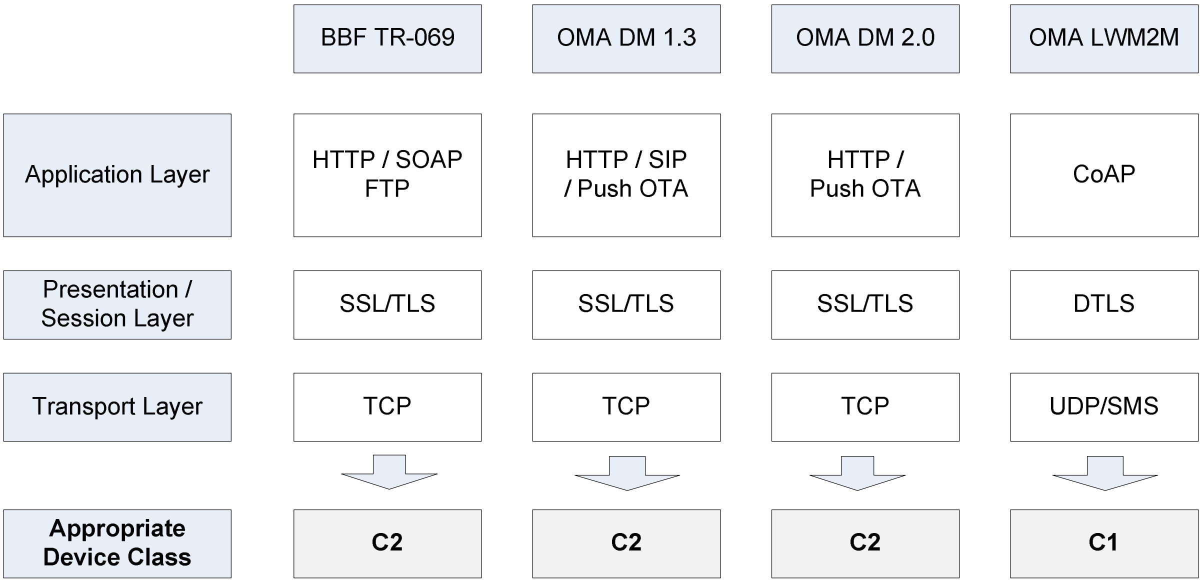
- Class 0 (C0)
  - Devices are very constrained (i.e. CPU, RAM, Flash) sensor-like nodes.
  - No possibility to have a direct communications with the Internet in a secure manner.
  - Deployed with a larger device acting as a management proxier and/or gateway.
- Class 1 (C1)
  - Has the capability to connect with nodes across the Internet in a secure manner using a constrained protocol stack (e.g. DTLS, UDP, CoAP) and various encoding protocols (e.g. TLV, JSON, Javascript).
  - Messages between nodes are typically transmitted within 1 packet due to the cost of packet fragmentation and reassembly within the Device.
  - Devices typically support one M2M Application.
  - Devices have simple mechanisms in place to communicate behind network firewalls and NATs.
- Class 2 (C2)
  - Has the capability to connect with nodes across the Internet in a secure manner using a full featured and reliable protocol stack that typically consists of TCP, HTTP, TLS (security) and various encoding protocols (e.g. XML, SOAP).
  - Devices have mechanisms in place to communicate behind network firewalls and NATs.
- Class 3 (C3)
  - Has the capability to be deployed as a management proxy - connecting C0 devices to nodes within the Internet.
  - Provides additional gateway features (e.g. multiple M2M applications).

NOTE 1: [i.25] defines only 3 classes from C0 to C2. In this annex, C3 is defined to designate some management proxier and/or gateway as a new class, higher than C2.

NOTE 2: The class of a device does not mean a device cannot be deployed with lightweight and energy-efficient aspects of the transport protocols which can consume less bandwidth across the network (e.g. HTTP compression).

A.2 Device Classes and Management Technologies

The existing management technologies utilize the different protocol stacks and the different protocol stacks consumes different amount of memory. The figure below demonstrates what kinds of protocol stacks are used by the management technologies and the appropriate device class of each technology based on the analysis of the previous clause A.1.



	BBF TR-069	OMA DM 1.3	OMA DM 2.0	OMA LWM2M
Application Layer	HTTP / SOAP FTP	HTTP / SIP / Push OTA	HTTP / Push OTA	CoAP
Presentation / Session Layer	SSL/TLS	SSL/TLS	SSL/TLS	DTLS
Transport Layer	TCP	TCP	TCP	UDP/SMS
Appropriate Device Class	**C2**	**C2**	**C2**	**C1**

Figure A.2.1: Protocol Stacks and Device Class used by the management technologies

It means that, as an instance, the BBF TR-069 enabled devices might be consisted of Embedded OS (2 KB) + TCP (4 KB) + SSL/TLS (36 KB) + HTTP (4 KB) + REST Engine (0,7 KB) + BBF TR-069 Client (less than 150 KB) < 250 KB (C2 Requirement) to fulfil the BBF TR-069's protocol and resource requirement for devices that do not support the TR-069 proxy features. For the OMA LWM2M enabled devices can be operated on Embedded OS (2 KB) + UDP (1,3 KB) + DTLS (36 KB) + CoAP (4 KB) + REST Engine (0,7 KB) + LWM2M Client (less than 20 KB) < 100 KB (C1 Requirement).

NOTE 1: [i.26] is referred to indicate the code size of each protocol.

NOTE 2: The size of Embedded OS is based on Contiki OS which is an OS for tiny networked sensors. The summation of size of all modules such as kernel, program loader, multi-threading and timer library is 2 280 bytes.

To sum up, this guideline identifies the minimum resource required on prospective oneM2M Device and Gateway by implementing different existing management technologies on resource constrained devices.

History

Document history		
V1.0.0	April 2015	Publication