

# ITU-T

TELECOMMUNICATION  
STANDARDIZATION SECTOR  
OF ITU

# Y.4500.14

(03/2018)

SERIES Y: GLOBAL INFORMATION  
INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS,  
NEXT-GENERATION NETWORKS, INTERNET OF  
THINGS AND SMART CITIES

Internet of things and smart cities and communities –  
Frameworks, architectures and protocols

---

## **oneM2M – LwM2M interworking**

Recommendation ITU-T Y.4500.14

## ITU-T Y-SERIES RECOMMENDATIONS

### GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

#### GLOBAL INFORMATION INFRASTRUCTURE

General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899

#### INTERNET PROTOCOL ASPECTS

General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999

#### NEXT GENERATION NETWORKS

Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999

#### FUTURE NETWORKS

Y.3000–Y.3499

#### CLOUD COMPUTING

Y.3500–Y.3999

#### INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES

General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
<b>Frameworks, architectures and protocols</b>	<b>Y.4400–Y.4549</b>
Services, applications, computation and data processing	Y.4550–Y.4699
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

*For further details, please refer to the list of ITU-T Recommendations.*

# Recommendation ITU-T Y.4500.14

## oneM2M – LwM2M interworking

### Summary

Recommendation ITU-T Y.4500.14 specifies the interworking capabilities of the machine to machine (M2M) service layer between application service node/infrastructure node/middle node (ASN/IN/MN) common services entities (CSEs) and lightweight machine to machine (LWM2M) endpoints.

### History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4500.14	2018-03-01	20	<a href="http://handle.itu.int/11.1002/1000/13509">11.1002/1000/13509</a>

### Keywords

Interworking, LwM2M, oneM2M.

---

\* To access the Recommendation, type the URL <http://handle.itu.int/> in the address field of your web browser, followed by the Recommendation's unique ID. For example, <http://handle.itu.int/11.1002/1000/11830-en>.

## FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

## NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

## INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

NOTE – This Recommendation departs slightly from the usual editorial style of ITU-T Recommendations to preserve existing cross-referencing from external documents.

© ITU 2018

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

## Table of Contents

	<b>Page</b>
1 Scope.....	1
2 References.....	1
3 Definitions .....	1
3.1 Terms defined elsewhere .....	1
3.2 Terms defined in this Recommendation .....	2
4 Abbreviations and acronyms .....	2
5 Conventions .....	2
6 Architecture model .....	3
6.1 Introduction .....	3
6.2 Reference model .....	3
6.3 Types of interworking .....	4
6.4 Composition of the interworking proxy entity .....	5
7 Architecture aspects.....	6
7.1 Introduction .....	6
7.2 LWM2M device and endpoint lifecycle.....	6
7.3 LWM2M object discovery .....	8
7.4 LWM2M object transport and interworking .....	11
7.5 LWM2M object subscription and notification .....	16
7.6 LWM2M object security .....	18
7.7 LWM2M IPE administration and maintenance.....	19
7.8 LWM2M client provisioning (Bootstrap) .....	20
8 Transparent interworking function .....	20
8.1 Introduction .....	20
8.2 Attribute mapping for the <contentInstance> resources .....	21
9 Semantically enabled interworking function (Informative) .....	21
9.1 Introduction .....	21
9.2 Organization of semantically enabled content sharing resources.....	21
9.3 Guidelines for mapping to the base ontology .....	23
Annex A – oneM2M specification update and maintenance control procedure.....	25
Appendix I – Introduction to OMA LightweightM2M (LWM2M).....	26
I.1 Introduction .....	26
I.2 Architecture .....	27
I.3 Terminology .....	27
I.4 Reference points .....	28
I.5 Protocols .....	29
I.6 Functions .....	33
Bibliography.....	34



# Recommendation ITU-T Y.4500.14

## oneM2M – LwM2M interworking

### 1 Scope

This Recommendation specifies the interworking capabilities of the machine to machine (M2M) service layer between application service node/infrastructure node/middle node (ASN/IN/MN) common services entities (CSEs) and lightweight machine to machine (LWM2M) endpoints using the architecture identified in Annex VI of oneM2M TS-0001 [ITU-T Y.4500.1] for the following interworking scenarios:

- Interworking for transparent transport of encoded LWM2M objects and commands in content sharing resources between LWM2M endpoints and M2M applications.
- Interworking with full mapping of LWM2M objects in LWM2M endpoints to semantically enabled content sharing resources that are utilized by M2M applications.

NOTE – The present Recommendation limits content sharing resources to <container> and <contentInstance> resources.

The Recommendation contains oneM2M Release 2 specification oneM2M LWM2M Interworking V2.0.0 and is equivalent to standards of oneM2M partners including ARIB, ATIS [b-ATIS.oneM2M.TS0014], CCSA [b-CCSA M2M-TS-0014], ETSI [b-ETSI TS 118 114], TTA, TSDSI [b-TSDSI STD T1.oneM2M TS-0014], TTA [b-TTAT.MM-TS.0014] and TTC [b-TTC TS-M2M-0014].

### 2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- |                   |  |
|-------------------|--|
| [ITU-T Y.4500.1]  | Recommendation ITU-T Y.4500.1 (2018), <i>one M2M - Functional architecture</i> .   |
| [ITU-T Y.4500.11] | Recommendation ITU-T Y.4500.11 (2018), <i>one M2M - Common terminology</i> .   |
| [ETSI TS 118 103] | ETSI TS 118 103 (2016), <i>one M2M; Security solutions (oneM2M TS-0003 version 2.4.1 Release 2)</i> .                                    |
| [OMA TS LwM2M]    | OMA-TS-LightweightM2M-V1-0-20150318-D, <i>Lightweight Machine to Machine Technical Specification, March 2015, Open Mobile Alliance</i> . |

### 3 Definitions

For the purposes of the present Recommendation, the terms and definitions given in oneM2M TS-0011 [ITU-T Y.4500.11] and oneM2M TS-0001 [ITU-T Y.4500.1] apply.

#### 3.1 Terms defined elsewhere

This Recommendation uses the following term(s) defined elsewhere:

**3.1.1 application entity** [ITU-T Y.4500.11]: Represents an instantiation of application logic for end-to-end M2M solutions.

**3.1.2 common services entity (CSE)** [ITU-T Y.4500.11]: Represents an instantiation of a set of common service functions of the M2M environments. Such service functions are exposed to other entities through reference points.

## **3.2 Terms defined in this Recommendation**

None.

## **4 Abbreviations and acronyms**

For the purposes of the present Recommendation, the abbreviations given in oneM2M TS-0011 [ITU-T Y.4500.11] and the following apply:

AE	Application Entity
ASN	Application Service Node
CoAP	Constrained Application Protocol
CSE	Common Services Entity
DTLS	Datagram, Transport Layer Security
IPE	Interworking Proxy application Entity
IN	Infrastructure Node
JSON	JavaScript Object Notation
LWM2M	Lightweight Machine to Machine
M2M	Machine to Machine
MCU	Multipoint Control Unit
MN	Middle Node
OMA	Open Mobile Alliance
RAM	Random Access Memory
SMS	Short Message Service
UDP	User Datagram Protocol

## **5 Conventions**

The keywords "shall", "shall not", "should", "should not", "may", "need not" in the present Recommendation are to be interpreted as described:

Shall/Shall not:

### **Requirements**

- 1) Effect on Standard: Standard needs to describe the required feature (i.e., specify a technical solution for the Requirement);
- 2) Effect on products: every implementation (M2M Solution that complies to this Standard) must support it;
- 3) Effect on deployments: every deployment (M2M Service based on this Standard) must use the Standardized feature where applicable – otherwise e.g., interoperability problems with other services could arise.



Should/Should not:

### Recommendation

- 1) Effect on Standard: Standard needs to describe a solution that allows the presence and the absence of the feature;
- 2) Effect on products: an implementation may or may not support it, however support is recommended;
- 3) Effect on deployments: a deployment may or may not use it, however usage is recommended.

May/Need not:

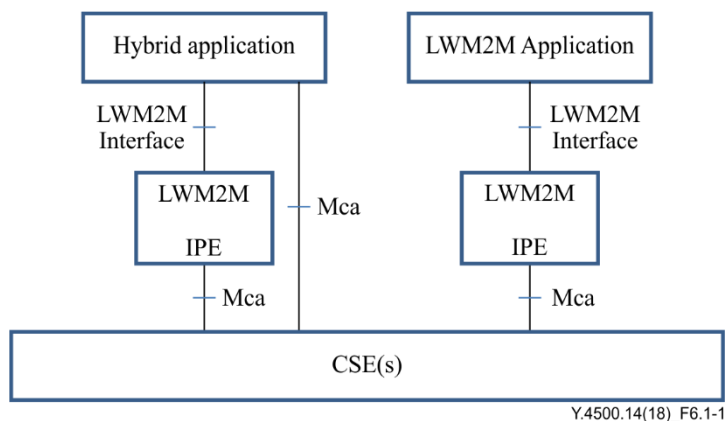
### Permission/Option

- 1) Effect on Standard: Standard needs to describe a solution that allows the presence and the absence of the required feature;
- 2) Effect on products: an implementation may or may not support it;
- 3) Effect on deployments: A deployment may or may not use it.

## 6 Architecture model

### 6.1 Introduction

The architecture model followed in the present Recommendation is based on the architecture model in Annex VI of oneM2M TS-0001 [ITU-T Y.4500.1] that describes how interworking between CSEs and non-oneM2M solutions and protocol using specialized interworking proxy application entities (IPEs). The present Recommendation describes the LWM2M IPE that supports the interworking scenarios depicted in Figure 6.1-1.

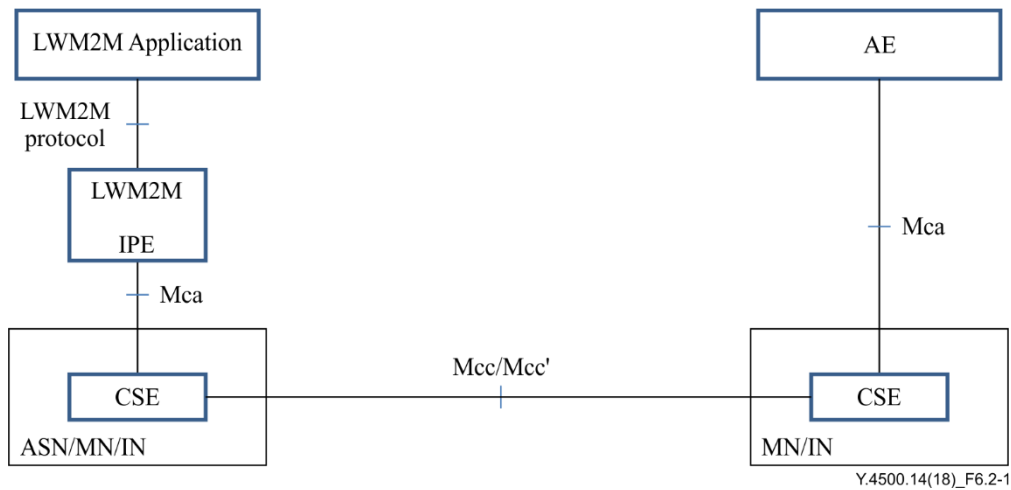


**Figure 6.1-1 – LWM2M interworking scenarios**

In the scenarios depicted in Figure 6.1-1, the hybrid and LWM2M applications represent applications that implement the LWM2M client role defined in the LWM2M protocol [OMA TS LwM2M].

### 6.2 Reference model

The LWM2M interworking reference model, shown in Figure 6.2-1, utilizes the functional architecture's reference model in oneM2M TS-0001 [ITU-T Y.4500.1]; augmenting the oneM2M TS-0001 [ITU-T Y.4500.1] reference model with capabilities provided by the LWM2M IPE.



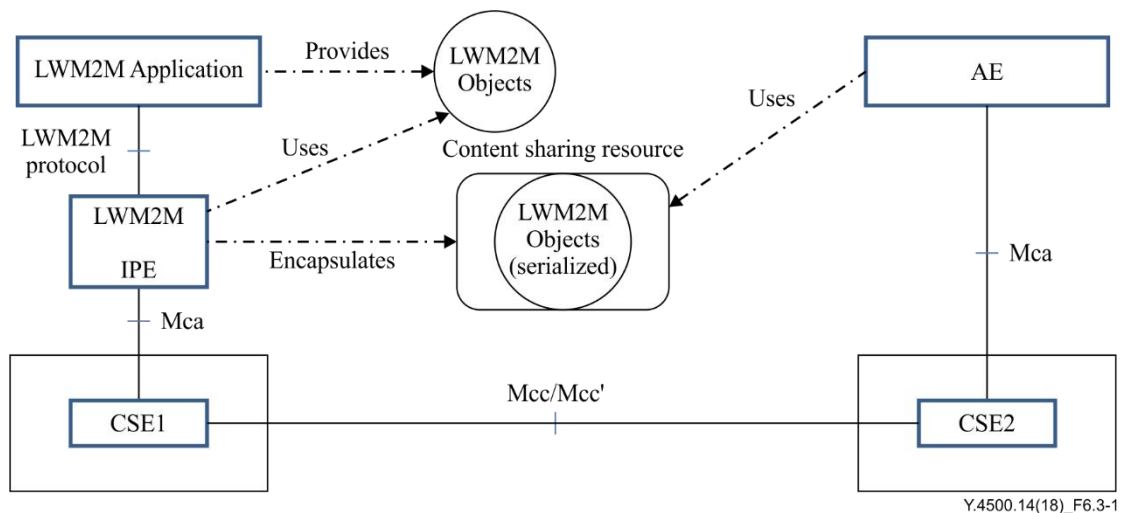
NOTE – The application entity (AE) in the reference model could be registered with the same CSE as the LWM2M IPE.

**Figure 6.2-1 – LWM2M reference model**

### 6.3 Types of interworking

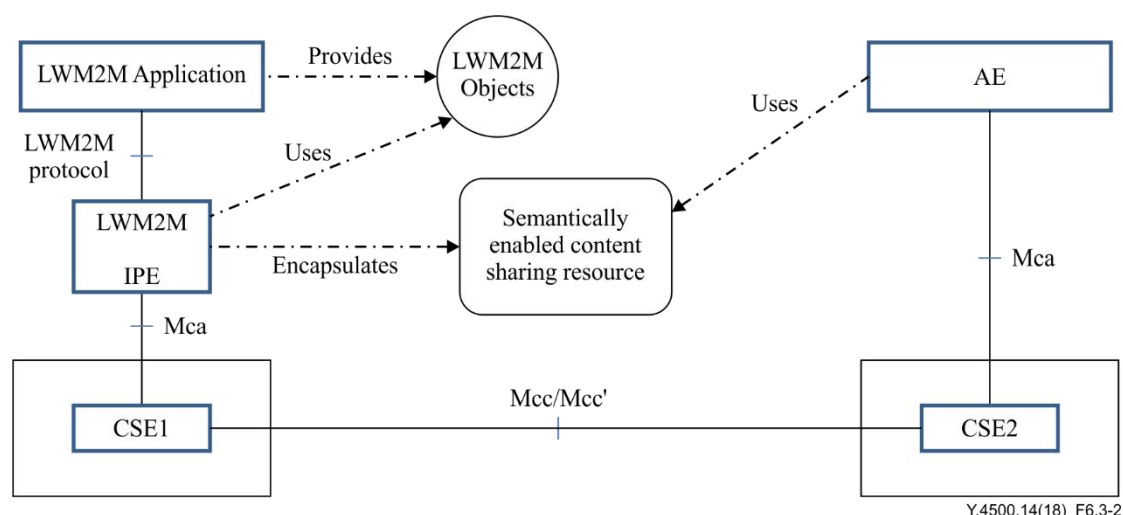
LWM2M IPEs provide the following types of interworking:

- 1) Interworking using the content sharing resource for transparent transport of encoded LWM2M objects that are available to application entities (AEs) as depicted in Figure 6.3-1.
- 2) Interworking with full mapping of the semantics of LWM2M objects to semantically enabled resources that are available to AEs as depicted in Figure 6.3-2.
- 3) While depicted outside the hosting CSE, the content sharing resources are hosted in a CSE (e.g., CSE1).



**Figure 6.3-1 – LWM2M transparent interworking function**

In Figure 6.3-1, the LWM2M objects are provided by the LWM2M application to the LWM2M IPE using the LWM2M protocol. The LWM2M IPE then encapsulates the LWM2M objects in content sharing resources and then hosts the content sharing resources in a CSE using the Mca reference points for use by AEs. The AE accesses the content sharing resource from the CSE that hosts the resource using the Mca reference point. Once the AE receives the content sharing resource, the AE extracts the LWM2M object from the content sharing resource for the AE's purpose. Clause 7 describes this type of interworking in greater detail.



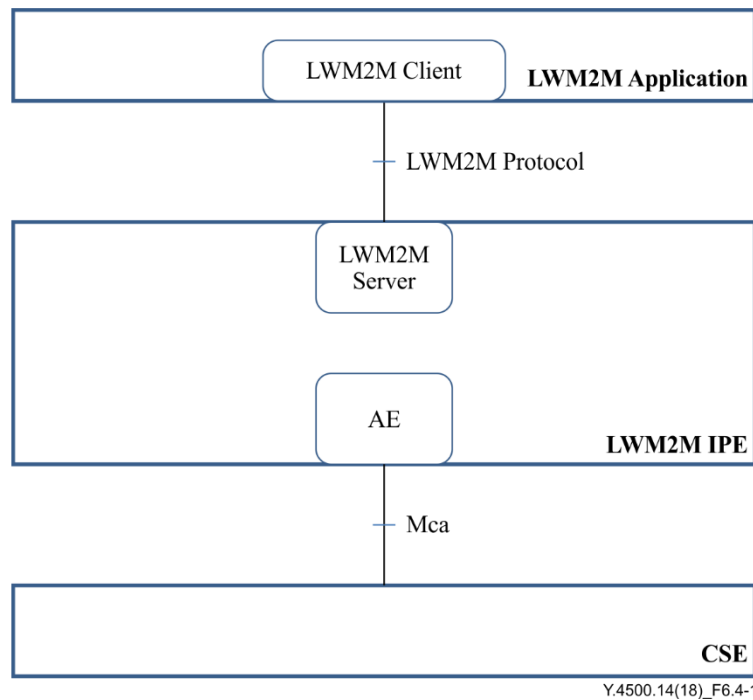
**Figure 6.3-2 – LWM2M semantically enabled interworking function**

In Figure 6.3-2, the LWM2M objects are provided by the LWM2M Application to the LWM2M IPE using the LWM2M Protocol. The LWM2M IPE then interworks the LWM2M objects into content sharing resources. The content sharing resources are based on the oneM2M defined Semantic Ontology. The LWM2M IPE hosts the content sharing resource in a CSE across the Mca reference for use by other AEs. The AE accesses the content sharing resource from the CSE that hosts the resource using the Mca reference point. Once the AE receives the content sharing resource, the AE encodes the information using the Semantic Ontology for the AE's purpose. Clause 9 describes this type of interworking in greater detail.

An instance of a LWM2M IPE shall provide the capability for transparent transport of encapsulated LWM2M objects as content sharing resources and/or translation of LWM2M objects as oneM2M semantically enabled content sharing resources.

## 6.4 Composition of the interworking proxy entity

The LWM2M IPE participation in the LWM2M protocol as described in clause 5 does so in the role of a LWM2M server to which LWM2M applications (LWM2M clients) interact. For each LWM2M client (endpoint) that is maintained by the LWM2M server in the LWM2M IPE, the LWM2M IPE shall instantiate and maintains an instance of a resource of type <AE>. Figure 6.4-1 shows LWM2M IPE architecture.



**Figure 6.4-1 – LWM2M IPE architecture**

## **7 Architecture aspects**

### **7.1 Introduction**

The LWM2M IPE participation in the LWM2M protocol as described in clause 5 does so in the role of a LWM2M server with which LWM2M applications (LWM2M clients) interact. As a LWM2M server, the IPE provides the following architecture aspects based on the LWM2M protocol aspects described in clause I.2:

- LWM2M device and endpoint lifecycle (client registration).
- LWM2M object discovery (client registration, device management and service enablement).
- LWM2M object transport and interworking (device management and service enablement).
- LWM2M object subscription and notification (information reporting).
- LWM2M interworking proxy entity administration.
- LWM2M client provisioning (bootstrap).
- LWM2M object security (device management and service enablement).

### **7.2 LWM2M device and endpoint lifecycle**

#### **7.2.1 Introduction**

As the LWM2M IPE discovers LWM2M endpoints when the LWM2M IPE interacts with the LWM2M client over the LWM2M protocol's bootstrap and client registration interfaces, the LWM2M IPE shall maintain the associated resources in the CSE that represents the LWM2M device and endpoint.

#### **7.2.2 LWM2M device and endpoint resource representation**

##### **7.2.2.1 Introduction**

LWM2M endpoint provides the management and control functions for an M2M application on a device. As such, the CSE that hosts the M2M application shall represent the LWM2M endpoint as a

<AE> resource (LWM2M endpoint <AE> resource). The LWM2M Device that hosts the LWM2M endpoint shall be represented as a <node> resource.

#### **7.2.2.2 LWM2M device and endpoint resource identification**

LWM2M endpoints are identified by their endpoint client name described in clause 7.2.1 of the LWM2M Technical Specification [OMA TS LwM2M]. The endpoint client name URN without the "urn:" sequence is used as the AE-ID of the associated <AE> resource that represents the LWM2M client.

In most deployment scenarios, LWM2M devices host one (1) LWM2M endpoint. In this scenario the LWM2M service's <node> resource's M2M-Node-ID should be the same as the LWM2M endpoint client name URN without the "urn:" sequence. When a LWM2M device hosts more than one (>1) LWM2M endpoint, the determination of the <node> resource's M2M-Node-ID is implementation specific. In all deployment scenarios, the <AE> resource is linked with the <node> resource as described in oneM2M TS-0001 [ITU-T Y.4500.1].

As the LWM2M endpoint is represented as an <AE> resource and a LWM2M object is represented as a content sharing resource in the M2M service layer, a reference shall be made between the <AE> resource that represents the LWM2M endpoint and the content sharing resources which represents the list of LWM2M objects and object instances available in that LWM2M client.

In order to identify interworked entities hosted in a CSE for the LWM2M technology described in this present Recommendation, the <AE> resource that represents the LWM2M endpoint and the content sharing resources which represent the list of LWM2M objects and object instances available in that LWM2M client, shall have a *Iwked\_Technology labels* attribute set to LWM2M.

In addition the <AE> resource uses the hierarchical and non-hierarchical mechanisms for resource addressing as defined in clause 9.3.1 of oneM2M TS-0001 [ITU-T Y.4500.1] where the *resourceName* attribute of the <AE> resource shall be an endpoint client name URN without the "urn:" sequence.

#### **7.2.2.3 LWM2M endpoint lifecycle**

LWM2M endpoint's are discovered when the LWM2M client is successfully registers with the LWM2M server using the LWM2M Register operation. In addition to the LWM2M register operation, the LWM2M client can periodically refresh the LWM2M client's registration with the LWM2M IPE using the LWM2M Update operation. Finally a LWM2M client can deregister when the LWM2M client issues a de-register operation to the LWM2M IPE or the LWM2M client's registration lifetime expires.

The LWM2M client registration interface's operations and the registration lifetime expiration event maps to the following operations on the <AE> and <node> resources.

**Table 7.2.2.3-1 – LWM2M endpoint lifecycle translation – Client registration interface**

<b>LWM2M Operation Client Registration Interface</b>	<b>oneM2M Resource and Operation</b>
Register	create <AE>, create <Node>
Update	update <AE>, update <Node>
De-register	delete <AE>, delete <Node>

**Table 7.2.2.3-2 – LWM2M endpoint lifecycle translation – LWM2M server events**

<b>LWM2M Server Events</b>	<b>oneM2M Resource and Operation</b>
client lifetime expiration	delete <AE>, delete <Node>, delete <container> resources associated with the <AE> resource

**Table 7.2.2.3-3 – LWM2M endpoint lifecycle attribute translation**

<b>LWM2M Attributes Client Registration Interface</b>	<b>oneM2M Resource Attribute</b>
Endpoint Client Name	<AE>: AE-ID, resourceName <Node>: M2M-Node-ID when the Device only supports one Endpoint; resourceName
Lifetime	<AE>, <Node>: expirationTime
LWM2M Version	<AE>, <Node>: labels. Value is "lwm2m-Entity-Version:" appended with the value of the LWM2M Version
Binding Mode	Not Applicable
SMS Number	Not Applicable

**Table 7.2.2.3-4 – LWM2M endpoint lifecycle response code translation**

<b>LWM2M Errors Client Registration Interface</b>	<b>oneM2M Resource Operation Response</b>
Register	create <AE>, create <Node>
2.01 Created:	2001 Created
4.00 Bad Request	All other codes
4.03 Forbidden	4105 Conflict
Update	update <AE>, update <Node>
2.04 Changed	2004 Changed
4.00 Bad Request	All other codes
4.04 Not Found	4000 Not Found
De-register	delete <AE>, delete <Node>
2.02 Deleted	2002 Deleted
4.04 Not Found	4004 Not Found

#### 7.2.2.4 Configuration of CMDH policies

In the present Recommendation, the CMDH Policies associated with the <Node> resource for the AE is implementation specific.

#### 7.2.2.5 Registering a registered LWM2M endpoint

In the scenario where a LWM2M client issues a register operation for an AE that is already created, the LWM2M IPE shall treat the operation as if the LWM2M client requested a de-Register (Delete <AE> resource) prior this register operation (Create <AE> resource) as described in Table 7.2.2.3-1. The procedure for the LWM2M server is defined in clause 6.3.1 of the LWM2M Technical Specification [OMA TS LwM2M].

### 7.3 LWM2M object discovery

#### 7.3.1 Introduction

The LWM2M client uses the registration interface to provide the properties required by the LWM2M server of the IPE to contact the LWM2M client (e.g., endpoint name) and to maintain the session between these two LWM2M entities (e.g., Lifetime, Queue Mode). In addition, the LWM2M

client also provides the knowledge of the supported objects and existing object instances across the registration Interface.

The LWM2M IPE uses the information exchange across this interface to synchronize which LWM2M objects supported by the LWM2M endpoint and what is defined in the hosting CSE for the M2M application representing the LWM2M endpoint. This clause specifies how discovered LWM2M objects identifiers are translated to discoverable content sharing resources along with the associated linkages to other resources.

### **7.3.2 LWM2M object identifier representation**

#### **7.3.2.1 Introduction**

Through the registration Interface, the LWM2M client provides the list of supported LWM2M objects and existing LWM2M object instances. Each element of the list is described by its path, which can be the path of an object or an object instance.

For example the LWM2M client could provide the following list of paths: `</1/0>`, `</1/1>`, `</2/0>`, `</2/1>`, `</3/0>`, `</4/0>`, `</5>`. This list of paths is a valid list of LWM2M objects and LWM2M object instances in the CoRE Link Format [RFC6690], specifying that LWM2M objects with OMNA Identifiers 1, 2, 3, 4 and 5 are supported. The respective OMNA references are: `urn:oma:lwm2m:oma:1`, `urn:oma:lwm2m:oma:2`, `urn:oma:lwm2m:oma:3`, `urn:oma:lwm2m:oma:4`, `urn:oma:lwm2m:oma:5`.

Additionally, information is provided that LWM2M object 1 (server object) and LWM2M object 2 (access control object) have 2 instances (instance identifiers 0 and 1); LWM2M object 3 (device object) and LWM2M object 4 (connectivity monitoring object) have 1 instance each (0); LWM2M object 5 is supported but no instance has been created yet for that LWM2M object.

Optionally other information can be carried by that list as the capability for all the objects in the LWM2M client to support:

- an alternate root path (default root path is `'/'`);
- a specific Content-Format (e.g., LWM2M JSON Content-Format).

For discovery of LWM2M objects by M2M applications, the properties carried by LWM2M objects list (i.e., technology, objects and LWM2M object instances identifiers, optional alternate rootpath, supported Content-Format) shall be translated into the labels attribute of the content sharing resource as separate entries with the following format:

- Iwked-Technology:LWM2M.
- Iwked-Entity-Type:Resource Type.
- LWM2M-PATH:Resource Root Path (for LWM2M default rootpath is `"/"`).
- Iwked-Entity-ID:Resource Path object identifier and instance identifier.
- Iwked-Content-Type: Supported content format (LWM2M default Supported ContentFormat is LWM2M TLV other can be LWM2M JSON).

For example if the LWM2M endpoint provided the following LWM2M objects as part of the client registration Interface: `</lwm2m>;rt="oma.lwm2m";ct=LWM2M+JSON,</1/0>` would translate into a `<container>` resource with the following entries in the labels attribute: Iwked-Technology:LWM2M Iwked-Entity-Type:"`urn:oma:lwm2m:oma:1`" LWM2MPATH:"`/lwm2m`" Iwked-Entity-ID: "`/1/0`" Iwked-Content-Type:LWM2M+JSON (see Note).

NOTE – LWM2M+JSON is an entry (numerical ID) in the CoAP Content-Format Registry representing the media-type "`application/vnd.oma.lwm2m+json`" used in LWM2M TS 1.0 enabler and currently engaged in the IANA registration process.

The CoAP resource Type may also be used as the semantic ontology of the `<container>` resource by inserting this value in the ontologyRef attribute of the `<container>` resource.

LWM2M object resources are identified by their URI within the context of the LWM2M endpoint described in clause 7.2.1 of the LWM2M Technical Specification [OMA TS LwM2M].

As the LWM2M endpoint is represented as an <AE> resource and a LWM2M object is represented as a content sharing resource in the M2M service layer, a reference shall be made between the <AE> resource that represents the LWM2M endpoint and the content sharing resources which represent the list of LWM2M objects and object instances available in the LWM2M client.

In addition, content sharing resources that represents the LWM2M object or LWM2M object instance uses the hierarchical and non-hierarchical mechanisms for resource addressing as defined in clause 9.3.1 of oneM2M TS-0001 [ITU-T Y.4500.1] where the *resourceName* attribute of the content sharing resource shall be the value of the LWM2MURI.

For example if the LWM2MURI is "/1/0" and the LWM2MPATH is "/" then the *resourceName* attribute of the content sharing resource could be "/1/0".

### 7.3.2.3 LWM2M object lifecycle

LWM2M endpoints are discovered when the LWM2M client successfully registers with the LWM2M server using the LWM2M register operation. In addition to the LWM2M register operation, the LWM2M client can periodically refresh the LWM2M client's registration with the LWM2M IPE using the LWM2M update operation. Finally a LWM2M client can de-register when the LWM2M client issues a de-register operation to the LWM2M IPE or the LWM2M client's registration lifetime expires.

The LWM2M client registration interface's operations and the registration lifetime expiration event maps to the following operations on the content sharing resource.

**Table 7.3.2.3-1 – LWM2M object lifecycle translation – Client registration interface**

LWM2M Operation Client Registration Interface	oneM2M Resource and Operation
Register	create <container>
Update	update <container>, delete <container>
De-register	delete <container>

**Table 7.3.2.3-2 – LWM2M object lifecycle translation – LWM2M server events**

LWM2M Server Events	oneM2M Resource and Operation
Client lifetime expiration	delete <container>

**Table 7.3.2.3-3 – LWM2M object lifecycle attribute translation**

LWM2M Attributes Client Registration Interface	oneM2M Resource Attribute
Endpoint Client Name	Not Applicable
Resource Links	<container> resourceName
Lifetime	<container> expirationTime
LWM2M Version	Not Applicable
Binding Mode	Not Applicable
SMS Number	Not Applicable



**Table 7.2.2.3-4 – LWM2M object lifecycle response code translation**

<b>LWM2M Errors Client Registration Interface</b>	<b>oneM2M Resource Operation Response</b>
Register 2.01 Created: 4.00 Bad Request 4.03 Forbidden	create <container> 2001 Created All other codes 4105 Conflict
Update 2.04 Changed 4.00 Bad Request 4.04 Not Found	update <container> 2004 Changed All other codes 4000 Not Found
De-register 2.02 Deleted 4.04 Not Found	delete <container> 2002 Deleted 4004 Not Found

## **7.4 LWM2M object transport and interworking**

### **7.4.1 Introduction**

When an oneM2M request is addressed from a CSE/AE to a hosting CSE containing the representation of a LWM2M client, the oneM2M response to the originator of the request is returned through the cooperation of the hosting CSE and the IPE.

The LWM2M client uses the device management and service enablement interface to provide the capabilities for the LWM2M server of the IPE to access LWM2M objects, objects instances and resources available from the LWM2M client.

A hosting CSE maintains a representation of the LWM2M data model of LWM2M object, object instance or resources as instances of oneM2M resource types. These oneM2M resources are instantiated and registered as described in clause 7.3 allowing oneM2M AEs and CSEs to exchange data with LWM2M clients.

In reference to clause 7.3, at the end of the registration phase all declared LWM2M object instances and LWM2M objects are associated to a content sharing resource created with the resourceName attribute set accordingly to the proper LWM2M object instance path (e.g., /9/1) or to the LWM2M object path (e.g., /9).

### **7.4.2 LWM2M interworking mechanisms**

#### **7.4.2.1 Introduction**

Cooperation between IPE and the oneM2M hosting CSE requires efficient mechanisms to maintain the latest state of the targeted LWM2M objects, object instances and resources. These mechanisms include data synchronization between the IPE and hosting CSE.

Data synchronization relies on the oneM2M subscription/notification and LWM2M observation/notification mechanisms. For automated data synchronization between the IPE and hosting CSE to be achieved, the solution shall be granular enough to allow data synchronization for each LWM2M object instance.

Access control mechanisms rely on an interworking between oneM2M and LWM2M access control policies.

LWM2M and oneM2M mechanisms used to achieve data synchronization and access control is specified in more detail in clauses 7.5 and 7.6.

These following sub-clauses specify the sequences of operations involved for each type of supported oneM2M requests following the general procedures specified in clause 10 (CREATE, RETRIEVE, DELETE) as used within the context of the interworking for the present Recommendation.

### 7.4.2.2 Relevant interworked resource settings

A LWM2M object instance is represented in oneM2M as a content sharing resource with 2 direct children resource types: a <subscription> resource and a <contentInstance> resource when used with a <container> resource.

For supporting the LWM2M interworking process, a few attributes for the Content Sharing resource and the <notification> resource shall have a specified set of parameters:

- a) Attributes of content sharing resource.

**Table 7.4.2.2-1 – <container> resource – Relevant interworked attributes**

Attributes of <container> resource	Value
<i>accessControlPolicyIDs</i>	ACP set (see clause 7.6)
<i>maxNrOfInstances</i>	1

- b) Child resource types of <container> resource.

**Table 7.4.2.2-2 – <container> resource – Relevant child resource types**

Child resources of <container> resource
<contentInstance> resource
<subscription> resource

- c) Attributes of <subscription> resource.

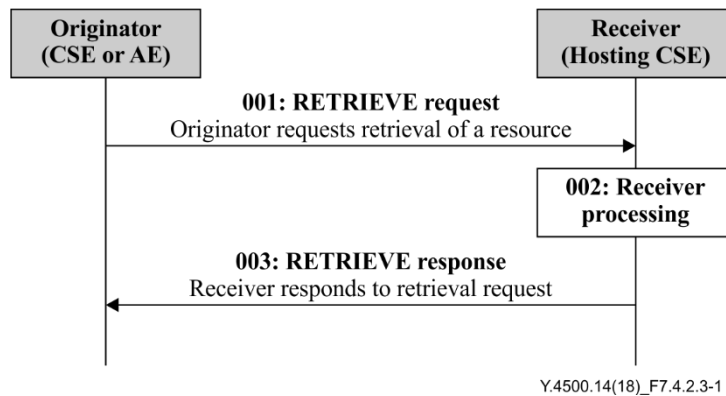
**Table 7.4.2.2-3 – <subscription> resource – Relevant interworked attributes**

Attributes of <subscription>	Description/Value
<i>notificationURI</i>	IPE URI
<i>eventType</i>	B. Deletion of the subscribed-to resource. C. Creation of a direct child of the subscribed-to resource. E. An attempt to retrieve a <contentInstance> direct-child-resource of a subscribed-to <container> resource is performed while this child is an obsolete resource or the reference used for retrieving this resource is not assigned. This retrieval is performed by a RETRIEVE request targeting the subscribed-to resource with the <i>Result Content</i> parameter set to either "child-resources" or "attributes+child-resources".

### 7.4.2.3 oneM2M RETRIEVE procedure

This procedure describes the retrieval of a resource using the oneM2M RETRIEVE request. The information contained within the resource is related to the LWM2M objects instances that are interworked through the IPE. This clause shall be treated in conformance with the RETRIEVE procedure specified in oneM2M TS-0001 [ITU-T Y.4500.1], clauses 10.1.2 and 10.2.4.2.

The receiver performs local processing to verify the existence of requested resource and checks privileges for retrieving the information related to the resource. After successful verification, the receiver shall return the requested information according to the procedures for the type of interworking (e.g., Transparent, Semantically Enabled) as described in clause 8 and 9, otherwise an error response shall be returned to the originator. Figure 7.4.2.3-1 shows the procedure for retrieving a resource.



**Figure 7.4.2.3-1 – Procedure for retrieving a resource  
(oneM2M TS-0001 [ITU Y.4500.1], clause 10)**

The target of the request is a <container> resource, the ResultContent and FilterCriteria parameters of the request (oneM2M TS-0001 [ITU-T Y.4500.1], clause 8.1.2) specify the nature of the information to retrieve.

**Table 7.4.2.3-1 – Effect of request parameters on retrieval request**

Request parameter: ResultContent	Request parameter: FilterCriteria	Effect
child-resources	resourceType = contentInstance	Retrieval of LWM2M object instance
attributes (default)	labels, attribute	Metadata retrieval of LWM2M object instance

Specific steps of the receiver processing according to the interworking process shall be followed:

**Step 001:** Find and verify the targeted <container> resource of the request: the resourceName corresponds to clause 7.3.2.

**Step 002:** Using the hosting CSE access control mechanisms, check for access control policy attached to the <container> resource of the request:

**Step 003:** On successful validation of the access control policy, proceed to fetch the requested information:

- a) If "ResultContent" is "attributes", jump to Step 4.
- b) If "ResultContent" is "child-resources" and "FilterCriteria" is "contentInstance", the associated <contentInstance> resource of the targeted container is considered.
  - **Step 003.1:** if the <contentInstance> resource is obsolete or its reference is not assigned, an event for Retrieval attempt (eventType 'E') is triggered to the entity that subscribed to the event (i.e., IPE); as a blocking procedure, the hosting CSE shall monitor the arrival of the new data or decide to report a timeout error in jumping to Step 004:
    - **Step 003.1.1:** On receiving the event of type 'E' (eventType 'E') the IPE performs a LWM2M READ request on the object instance of the targeted LWM2M client.
    - **Step 003.1.2:** Once the targeted object instance is available to IPE, the IPE creates and populates with that data the <contentInstance> child-resource of the requested <container> resource.
  - **Step 003.2:** The up-to-date information is available in the <contentInstance> resource.

**Step 004:** The hosting CSE returns the appropriate response back to the originator (e.g., errors, or data).

NOTE – As an OBSERVATION has been set up on the targeted LWM2M object instance, the automatic synchronization between the object instance and its representation in the hosting CSE is performed. Further oneM2M accesses to the resource should be simplified in minimizing impact of Step003 (up-to-date data already present from the Hosting CSE resources).

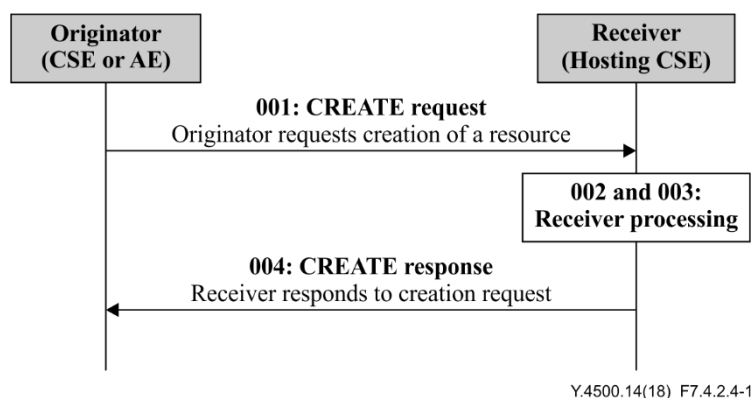
### General exceptions

- 1) The targeted resource/attribute in *To* parameter does not exist. The receiver shall respond with an error.
- 2) The originator does not have privileges to retrieve information stored in the resource on the receiver. The receiver shall respond with an error.
- 3) A timer has expired. The receiver shall respond with an error.

### 7.4.2.4 oneM2M CREATE procedure

This procedure describes the update of a LWM2M object instance in a LWM2M client using the oneM2M CREATE request. The information contained in the request via the content parameter (oneM2M TS-0001 [ITU-T Y.4500.1], clause 8.2.1) will be used to update an object instance in the LWM2M context.

This clause shall be treated in conformance with the CREATE Procedure specified in oneM2M TS-0001 [ITU-T Y.4500.1], clauses 10.1.2 and 10.2.4.1. Figure 7.4.2.4-1 shows the procedure for creating a resource.



**Figure 7.4.2.4-1 – Procedure for creating a resource  
(oneM2M TS-0001 [ITU-T Y.4500.1], clause 10)**

The target of the request is a <container> resource, the ResourceType parameters of the request (oneM2M TS-0001 [ITU-T Y.4500.1], clause 8.1.2) specifies the type of the resource to create.

**Table 7.4.2.4-1 – Effect of request parameters on create request**

Request parameter :ResourceType	Effect
ContentInstance	Replacement of the <latest> resource by a new one

In LWM2M interworking context, if there is already an existing <contentInstance> resource, creating a new one shall violates the policy defined in the parent <container> resource regarding the maxNrOfInstances which shall be set to 1. Then the oldest <contentInstance> resources shall be removed from the <container> to enable the creation of the new <contentInstance> resource.

In any case, a notification is sent to the IPE, which subscribed to the parent <container> resource with the eventType 'C' (Creation of a direct child of the subscribed-to resource). The IPE shall use that notification to update the LWM2M resource model with the new data received ("Content" parameter of the request).

Specific steps of the receiver processing according to the interworking process shall be as followed:

**Step 001:** Find and verify the <container> resource of the request: the resourceName corresponds to clause 7.3.2.

**Step 002:** Using the Hosting CSE Access Control mechanisms, check for access control policy attached to the <container> resource of the request.

**Step 003:** On unsuccessful validation of the access control policy, jump to step 4:

- **Step 003.1:** When according to the request, a <contentInstance> resource is created, an event for Child Creation (eventType 'C') is triggered to the entity that subscribed to that event (i.e., IPE).
- **Step 003.2:** On receiving the event of type 'C' the IPE - via the Mca reference point - get the data from the created <contentInstance> resource and propagates the updated data to the related object instance in the LWM2M client.

**Step 004:** The hosting CSE returns the appropriate response back to the originator (e.g., acknowledgment, errors).

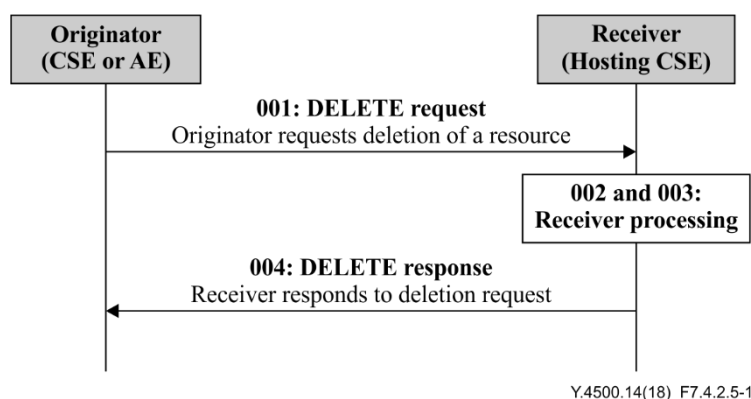
**General exceptions:**

- 1) The originator does not have the privileges to create a resource on the receiver, the Receiver shall respond with an error.
- 2) The resource with the specified name (if provided) already exists at the receiver, the Receiver shall response with an error.
- 3) The provided information in Content is not accepted by the receiver (e.g., missing mandatory parameter), the Receiver shall respond with an error.

#### 7.4.2.5 oneM2M DELETE procedure

This procedure describes the removal of a LWM2M object instance within a LWM2M client using the oneM2M DELETE request.

This clause shall be treated in conformance with the DELETE Procedure specified in oneM2M TS-0001 [ITU-T Y.4500.1], clauses 10.1.4 and 10.2.4.4. Figure 7.4.2.5-1 shows the procedure for deleting a resource.



**Figure 7.4.2.5-1 – Procedure for deleting a resource  
(oneM2M TS-0001 [ITU-T Y.4500.1], clause 10)**

Specific steps of the receiver processing according to the interworking process shall be as follows:

**Step 001:** Find and verify the <container> resource of the request: the resourceName corresponds to clause 7.3.2.

**Step 002:** Using the hosting CSE access control mechanisms, check for access control policy attached to the <container> resource of the request.

**Step 003:** On unsuccessful validation of the access control policy, jump to step 4:

- **Step 003.1:** When the <container> resource is deleted an event (eventType 'B': Deletion of subscribed-to resource) is triggered to the Entity that subscribed to that event (i.e., IPE).
- **Step 003.2:** On receiving the event type 'B', the IPE requests the LWM2M client to delete the object instance related to the <container> resource.
- **Step 003.3:** On object instance deletion, the LWM2M client performs a de-registration operation to the IPE.
- **Step 003.4:** Via the Mca reference point, the IPE communicates to the Hosting CSE the status of this de-registration.

**Step 004:** The hosting CSE returns the appropriate response back to the originator (e.g., Success, Errors).

### 7.4.3 oneM2M resource operation responses

**Table 7.4.3-1 – LWM2M response codes to oneM2M resource operation response codes**

<b>LWM2M Client Response Codes on device management and service enablement interface</b>	<b>oneM2M Resource Operation Response</b>
Create 2.01 Created: 4.00 Bad Request 4.01 Unauthorized 4.04 Not Found	2001 Created 4000 4103 4004
Read 2.05 Content: 4.01 Unauthorized 4.04 Not Found 4.05 Method Not Allowed	2000 OK 4103 4004 4005
Write 2.04 Changed 4.00 Bad Request 4.01 Unauthorized 4.04 Not Found 4.05 Method Not Allowed	2004 Changed 4000 4103 4004 4005 4102
Delete 2.02 Deleted 4.00 Bad Request 4.01 Unauthorized 4.04 Not Found 4.05 Method Not Allowed	2002 deleted 4103 4103 4004 4005

## 7.5 LWM2M object subscription and notification

### 7.5.1 Introduction

The LWM2M server uses the Information Reporting Interface to provide the capabilities for a LWM2M server to subscribe to changes to the LWM2M objects, LWM2M object instances and the associated LWM2M object's resources. Likewise the LWM2M client uses the Information Reporting Interface to notify subscribed LWM2M server's when the LWM2M object, LWM2M object instance and/or LWM2M object's resources change and to cancel the subscription on LWM2M objects, LWM2M object instances and the associated LWM2M object's resources.

The LWM2M server uses the device management and service enablement Interface to set the notification criteria for a subscription.

The oneM2M Subscription capabilities permit subscription changes to an oneM2M resource's attributes and its direct child resources. Likewise, the oneM2M Notification capabilities include a rich set of criteria for when a subscribed-to oneM2M resource is notified of a change.

### 7.5.2 LWM2M subscription procedure

The LWM2M IPE interworks the oneM2M resource's <subscription> child resource with the corresponding LWM2M object using the oneM2M <subscription> resource's attributes and the corresponding LWM2M object resource's Notification class Attributes.

NOTE – Each LWM2M object resource has an associated set of Notification class Attributes that are used for defining the applicable subscription and notification criteria.

When the LWM2M IPE creates a oneM2M content sharing resource, the LWM2M IPE creates a subscription on the content sharing resource to be notified whenever the oneM2M resource's subscription attribute is changed by setting the <subscription> resource's attributes as follows:

**Table 7.5.2-1 – LWM2M subscription procedure – <subscription> resource**

Attributes of <subscription>	Description
<i>accessControlPolicyIDs</i>	Link a <accessControlPolicy> resource with the privileges: accessControlOriginator originatorID set to the LWM2M IPE AE's AE-ID accessControlOperations: Set to RETRIEVE, CREATE, UPDATE, DELETE, DISCOVER, NOTIFY
<i>pendingNotification</i>	Set to "sendLatest"
<i>latestNotify</i>	Set to "latest"
<i>notificationContentType</i>	Set to "resource"
<schedule>	Set to immediate notification

Whenever another AE or CSE creates or deletes a subscription to the <container> resource, the LWM2M IPE shall be notified of the change and shall perform the following steps:

**Step 001:** Find the associated LWM2M object or object instance for notification's subscriptionReference.

**Step 002:** If the oneM2M notification indicates a subscription deletion:

- **Step 002a:** If the associated LWM2M object or object instance has an outstanding observation request from the LWM2M IPE then issue the LWM2M cancel observation operation.

**Step 003:** If the oneM2M notification indicates a subscription creation:

- **Step 003a:** If the associated LWM2M object or object instance does not have an outstanding Observation request from the LWM2M IPE then:
  - **Step 003a001:** Retrieve the parent resource of the <subscription> resource from the notification's subscriptionReference.
  - **Step 003a002:** Determine the LWM2M notification class attributes to set from the set of subscriptions of the parent resource using the <schedule> resource associated with each of the parent resource's subscriptions.
  - **Step 003a003:** Issue the LWM2M observe operation with the LWM2M notification class attributes.

### General exceptions

The processing for recovery of a failed LWM2M cancel observation or observation operation is vendor specific.

**Table 7.5.2-2 – LWM2M subscription procedure - Information reporting interface**

LWM2M Operation Information Reporting Interface	oneM2M Resource and Operation
Observe	NOTIFY (m2m:notification subscriptionDeletion=false)
Cancel Observation	NOTIFY (m2m:notification subscriptionDeletion=true)

**Table 7.5.2-3 – LWM2M subscription procedure attribute translation**

LWM2M Operation DM and SE Interface Notification class Attributes	oneM2M <subscription> Attribute
Minimum Period	<schedule> resource
Maximum Period	<schedule> resource
Greater Than	Not Applicable
Less Than	Not Applicable
Step	Not Applicable

### 7.5.3 LWM2M notification procedure

When the LWM2M IPE is notified by the LWM2M client of a change in a LWM2M object or object instance the LWM2M IPE creates a new <contentInstance> for the associated <container> resource according to the procedures for the type of interworking (e.g., Transparent, Semantic) as described in clauses 8 or 9.

## 7.6 LWM2M object security

### 7.6.1 Introduction

OMA-LWM2M and oneM2M access control policies shall collaborate in order to assure the interworked resources are accessible according to the oneM2M authorization procedure specified in clause 11.3.4 (M2M Authorization Procedure) of oneM2M TS-0001 [ITU-T Y.4500.1] and clause 7 (Authorization) of oneM2M TS-0003 [ETSI TS 118 103].

### 7.6.2 LWM2M interworking access control policy

The oneM2M access control policy mechanisms specified in clause 7 of oneM2M TS-0003 [ETSI TS 118 103], shall be used to check and validate the parameters of a request message against the ACPs (<accessControlPolicy> resources) which have been assigned to the accessed resource.

In order to assure a proper LWM2M interworking with oneM2M, the IPE shall setup the hosting CSE by:

- 1) providing a mandatory set of <accessControlPolicy> (ACPs) resources;
- 2) assigning a proper set of ACPs to the accessControlPolicyIDs attribute of each <container> resource allocated during the CSE registration phase (clause 7.3 LWM2M object Discovery).

The process for provisioning the IPE in order to perform such a setup is described in clause 7.6.3 "IPE and object security provisioning" of the present Recommendation.

In addition, the access control policy mechanisms specified in clause 7 of oneM2M TS-0003 [ETSI TS 118 103] are fully applicable in this LWM2M interworking context.

### 7.6.3 IPE and object security provisioning

In order to provide oneM2M information specified in the clause 7.6.2 (set of <accessControlPolicy> (ACPs) resources, assignment of accessControlPolicyIDs), the LWM2M IPE shall be supplied by information such as:

- a list of oneM2M originators and their associated access control rules likely to be exercised on the hosting CSE resources;



- a list of oneM2M originators likely to contact the LWM2M clients with the associated set of authorized operations.

In combining such an information with the access control policy specified in a given LWM2M client, see clause 6.8 "LWM2M client provisioning (Bootstrap)", the LWM2M IPE shall be able to provide to the hosting CSE, the oneM2M access control policy materials needed for properly registering LWM2M objects representation. In the current release of this Specification, this procedure of how the access control policy materials are provided is implementation specific.

## **7.7 LWM2M IPE administration and maintenance**

### **7.7.1 Introduction**

The IPE described in clause 6.4 is comprised functionality that includes the LWM2M server and the IPE's AE. This clause describes the administration and maintenance of these functional elements.

### **7.7.2 Administration and maintenance of the LWM2M server functionality**

#### **7.7.2.1 Introduction**

The LWM2M IPE provides the functionality that plays the role of the LWM2M server in order to communicate with LWM2M clients.

In order for communication to be established information needs to be provisioned into the LWM2M client and LWM2M server where the following artefacts are necessary to be established for the LWM2M server:

- LWM2M server and client credentials.
- LWM2M access control lists.

In addition, the LWM2M server maintains information about each LWM2M client and has actions that are used to maintain the LWM2M server.

These aspects of the LWM2M server are further described in clause E.2 of the LWM2M server resource [OMA TS LwM2M].

The mechanisms used to administer and maintain the LWM2M server functionality within the LWM2M IPE is out of scope of the present Recommendation.

#### **7.7.2.2 LWM2M server maintenance**

The LWM2M server maintains a set of LWM2M server account information for each LWM2M client that allows the LWM2M server to access and communicate with LWM2M client. These are:

- LWM2M server identifier associated with and assigned by the LWM2M client (LWM2M server identifier, registration lifetime).
- Policies for default observation behaviour.

The following actions can also be performed that affects the state of the LWM2M server's interaction with the LWM2M client:

- On-demand request for the LWM2M client to update its registration.

### **7.7.3 Maintenance of the LWM2M IPE AE context**

#### **7.7.3.1 Introduction**

The LWM2M IPE AE maintains information related to the operational context of the LWM2M IPE AE. Specifically the following elements are maintained for the LWM2M IPE AE:

- List of currently registered LWM2M endpoints.
- Configuration of the interworking functions to be used for the LWM2M objects and object instances.

### 7.7.3.2 LWM2M endpoint List

Whenever an LWM2M endpoint <AE> resource is created, updated or deleted as described in clause 7.2, the LWM2M IPE also manages the list of LWM2M endpoint <AE> resources using a oneM2M <group> resource.

The oneM2M <group> resource's lifecycle is linked to the LWM2M IPE <AE> resource's lifecycle.

**Table 7.7.3.2-1 – LWM2M IPE <AE> resource – Group lifecycle**

LWM2M IPE <AE> resource Operation	oneM2M Resource and Operation
create	create <group>. The group resourceName is the AE-ID of the LWM2M IPE <AE>.resource
update	update <group>
delete	delete <group>

The LWM2M endpoint <AE> resources' lifecycle operation maps to the following operations on the oneM2M <group> resource.

**Table 7.7.3.2-2 – LWM2M endpoint <AE> resource – Group member lifecycle**

LWM2M Endpoint <AE> resource Operation	oneM2M Resource and Operation
Create	update <group> (add member)
Delete	update <group> (delete member)

### 7.7.3.3 Configuration of interworking functions

Clause 6.3 describes the types of interworking functions as transparent interworking function and semantically enabled interworking function. An IPE provides the capability to perform one or both types of interworking functions. The granularity (e.g., object/object instance, IPE) that is used to define which interworking function(s) to use is implementation specific.

## 7.8 LWM2M client provisioning (Bootstrap)

This present Recommendation makes assumptions that the LWM2M clients and LWM2M servers functionality of the IPE have been provisioned with the proper LWM2M credential materials in accordance to the LWM2M specification (clause I.5.3.1 "Bootstrap") in order to securely communicate between the LWM2M client and LWM2M server.

Additionally, a LWM2M client connected to a LWM2M IPE, should be provisioned with the LWM2M access control policy information associated to the object instances contained in the LWM2M client as described in clause 7.6 "LWM2M object security". Additionally, in current release of this present Recommendation, the LWM2M server role of the LWM2M IPE does not contain the LWM2M Bootstrap server capability, consequently the LWM2M client provisioning operations shall be part of an out-of-band process.

## 8 Transparent interworking function

### 8.1 Introduction

Clause 6.3 introduced the transparent interworking function as depicted in Figure 6.3-1. This clause specifies the mappings of the attributes of the <contentInstance> resource for a <container> resource in order to allow an AE that uses the content sharing resource to understand that the content sharing resource has an encapsulated LWM2M object or object instance.

## 8.2 Attribute mapping for the <contentInstance> resources

When an AE accesses a <contentInstance> resource, the AE needs to know that the <contentInstance> resource encapsulates a LWM2M object or object instance as well as how the LWM2M object or object instance is encoded.

**Table 8.2-1 – Transparent interworking function mapping**

Interworking Function Mapping	oneM2M Resource Attribute
Indication that a LWM2M object or object instance is encapsulated in the <contentInstance> resource with the content type and encoding of the LWM2M object or object instance.	<contentInstance> resource: labels. Value is "LWM2M-Object-Encapsulation"
The content type of the LWM2M object or object instance based on the Content-Type option	<contentInstance>: contentInfo. Possible contentInfo values are translated from the LWM2M Content-Type option (see note).
NOTE: The LWM2M Technical Specification [OMA TS Lwm2M] defines the value to be used for the [encoding] if the Content-Type option is not present.	

## 9 Semantically enabled interworking function (Informative)

### 9.1 Introduction

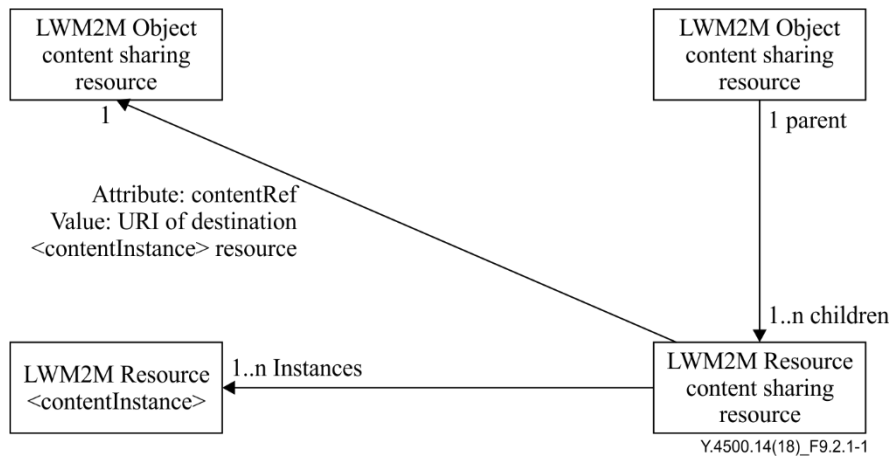
Clause 6.3 introduced the semantically enabled interworking function as depicted in Figure 6.3-2. This clause specifies how LWM2M objects and their associated LWM2M resources are organized as <container> resources in order for values associated with the LWM2M resources be translated into <contentInstance> resources. In addition, this clause specifies the mapping of content sharing resources the oneM2M Base Ontology [b-oneM2M TS-0012].

### 9.2 Organization of semantically enabled content sharing resources

#### 9.2.1 Introduction

Semantically enabled content sharing resources represent the structure and content of LWM2M objects and object instances by translating LWM2M objects, object instances and their LWM2M resources and LWM2M resource instances into a hierarchy of content sharing resources using the content sharing resource's parent-child relationship described in oneM2M TS-0001 [ITU-T Y.4500.1]. In addition, the LWM2M resources values are contained within the <contentInstance> resource for <container> resources.

When the LWM2M resource is of type LWM2M object Link, the <contentInstance> resource that represents the LWM2M resource is used to represent the LWM2M object Link by assigning the destination of the LWM2M object Link reference to another LWM2M object's content sharing resource. The reference is assigned using the <contentInstance> resource's contentRef attribute where the name of the attribute is "ObjectLink" and the value of the attribute is the URI of the destination content sharing resource. Figure 9.2.1-1 shows relationships of LWM2M semantically enabled content sharing resources.



**Figure 9.2.1-1 – Relationships of LWM2M semantically enabled content sharing resources**

### 9.2.2 Lifecycle of semantically enabled content sharing resources

Clauses 7.3 and 7.4 describe how LWM2M objects and object instances are discovered and instantiated. The semantic interworking function uses these procedures for instantiation of the content sharing resource for the LWM2M objects and object instances.

The content sharing resources for LWM2M resources and resource instances are created as child resources of the parent content sharing resource when the LWM2M object and object instance are created. Likewise these child content sharing resources are deleted when the parent content sharing resource is deleted.

Creation, update or deletion of one or more <contentInstance> resources for the LWM2M resource or resource instances that are not caused by the creation or deletion of the parent LWM2M object or object instance content sharing resource maps to the following operations on the LWM2M client.

**Table 9.2.2-1 – LWM2M resource content sharing resource lifecycle translation**

LWM2M Operation device management and service enablement Interface	oneM2M Resource and Operation LWM2M Resource or Resource instance Child Content Sharing Resource
Write	create child Content Sharing resource for new resource or resource instance. The name of the Content Sharing resource shall be the resource Id for a singleton LWM2M resource. The name of the Content Sharing resource shall be the LWM2M resource ID and LWM2M resource instance Id  When the LWM2M resource is of type LWM2M object Link, the <contentInstance> resource that represents the LWM2M resource is used to represent the LWM2M object Link by assigning the destination of the LWM2M object Link reference to another LWM2M object's Content Sharing resource. The reference is assigned using the <contentInstance> resource's contentRef attribute where the name of the attribute is "ObjectLink" and the value of the attribute is the URI of the destination <contentInstance> resource
Not applicable	update Content Sharing resource
Write	delete Content Sharing resource for resource or resource instance
Not Applicable	read Content Sharing resource

**Table 9.2.2-2 – LWM2M resource <contentInstance> lifecycle translation**

LWM2M Operation device management and service enablement Interface	oneM2M Resource and Operation LWM2M Resource or Resource instance <contentInstance> resource
Write	create <contentInstance>
Write - Sets the resource to default value	delete <contentInstance>
Read	read <contentInstance>

### 9.2.3 Mapping for the encoding of the <contentInstance> resource

When an AE accesses a <contentInstance> resource, the AE needs to know how the value of the resource or resource instance is encoded.

**Table 9.2.3-1 – Mapping of resource or resource instance encoding**

Interworking Function Mapping	oneM2M Resource Attribute
The encoding of the LWM2M Resource or resource instance based on the Content-Type option	<contentInstance>: contentInfo. Possible contentInfo values are translated from the LWM2M Content- Type option. Note: The LWM2M Technical Specification [OMA TS Lwm2M] defines the value to be used for the [encoding] if the Content- Type option is not present.

## 9.3 Guidelines for mapping to the base ontology

### 9.3.1 Introduction

Clause 9.2 describes the structure and relationships of the LWM2M objects and object instances along with their associated resources and resource instances. Using that structure this clause provides guidance on mapping the base ontology described by [b-oneM2M TS-0012] onto that resource structure. As ontologies are created for specific applications of LWM2M objects (e.g., Device management, Home automation), this clause can only be used for a basis of creating the application specific ontology because certain elements of base ontology (e.g., Aspects, Functionality, Services) cannot be inferred by the LWM2M definitions of LWM2M objects, object instances, resources or resource instances.

### 9.3.2 Mapping of the LWM2M client

LWM2M clients are represented as <AE> resources and are mapped to an InterworkedDevice. <AE> resources exposed by the LWM2M server associated with the IPE are mapped to the same area network.

### 9.3.4 Mapping of the LWM2M object, object instance, resource and resource instance

#### 9.3.4.1 Introduction

Mapping the LWM2M object, object instance, resource and resource instance to the base ontology is based on the following guidelines:

- LWM2M clients are mapped to InterworkedDevices.
- LWM2M objects are mapped to services and functionality.
- LWM2M resources that represent static (configured) properties of LWM2M objects are mapped to ThingProperties.
- LWM2M operation of execute map to operation and command.
- LWM2M Create, Update, Retrieve and Delete operations permitted for the LWM2M object or object instances map to operation and command.

- LWM2M resources (including those of type object Link) are mapped to Input-/OutputDataPoints:
  - Sub-structures of LWM2M resources are mapped to Variables that are sub-structures of Input-/OutputDataPoints (via the hasSubStructure relation):
    - Read-only LWM2M resources map to output DataPoint.
    - Write-only LWM2M resources map to input DataPoint.
    - Read-write LWM2M resources map to input DataPoint and output DataPoint with the same instance of a variable.
  - If the LWM2M object does not have a command state, the IPE will instantiate and maintain a <container> resource for command's state. In both instances the LWM2M resource that represents the command state maps to Output DataPoint.

## **Annex A**

### **oneM2M specification update and maintenance control procedure**

(This annex forms an integral part of this Recommendation.)

The provisions of Annex L in [ITU-T Y.4500.1] regarding oneM2M specification update and maintenance control procedure shall apply to this Recommendation.

## Appendix I

### Introduction to OMA LightweightM2M (LWM2M)

(This appendix does not form an integral part of this Recommendation.)

#### I.1 Introduction

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 user datagram protocol (UDP) and short message service (SMS) bearers.

NOTE – This appendix provides an overview of the LWM2M protocol. The authoritative source for the protocol is provided by the LWM2M Technical Specification [OMA TS LwM2M].

The crucial aspects in this work are the:

- Target devices for this protocol are resource constraint devices for example 8-16bit multipoint control unit (MCU), random access memory (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:

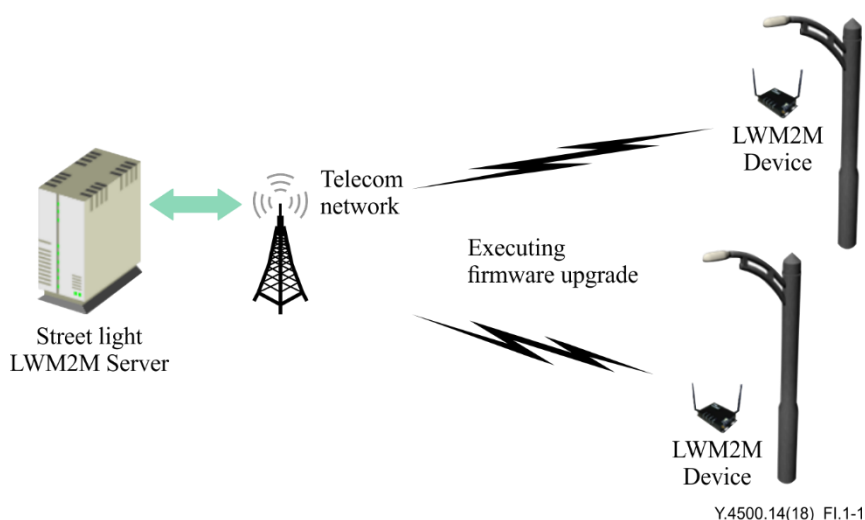
- Suitable for resource constraint devices.
- Usage of compact binary packets.
- Support for multiple data encoding formats that include binary, JavaScript object notation (JSON), plain text and opaque data formats.
- Support for reporting information from the server to the client when specified condition are met.
- Easy to be implemented though the reuse of existing implementation of IETF technologies: e.g., constrained application protocol (CoAP).
- CoAP for the transfer protocol and datagram, transport layer security (DTLS) [b-IETF RFC 6347] for securing the server/client exchanges.

One of the typical use cases of using LWM2M technology is the firmware upgrade of streetlights [b-OMA OMA-RD-LwM2M]:

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

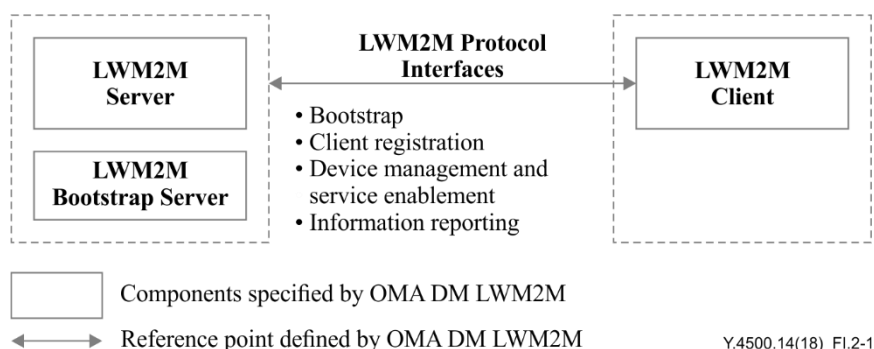
Figure I.1-1 shows firmware upgrade of streetlight from a use case using LWM2M





**Figure I.1-1 – Firmware upgrade of streetlight from a use case using LWM2M**

## I.2 Architecture



**Figure I.2-1 – LWM2M architecture**

As shown in the Figure I.2-1, the layout is the architecture of LWM2M [OMA TS LwM2M]. 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.

## I.3 Terminology

LWM2M [OMA TS LwM2M] is a RESTful protocol with concepts that are similar to oneM2M, however LWM2M uses different terms for these concepts. Table I.3-1 provides a comparison of applicable LWM2M and oneM2M terminology.

**Table I.3-1 – LWM2M/oneM2M terminology mapping**

LWM2M terminology	oneM2M terminology
Client Endpoint	<AE> resources that reside on devices and oneM2M nodes.
object, object instance	Resource in general; <contentInstance> resource when used for interworking.
Resource	Attribute for a resource.

## **I.4 Reference points**

### **I.4.1 Introduction**

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

### **I.4.2 Functional components**

#### **I.4.2.1 LWM2M server**

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

#### **I.4.2.2 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.

### **I.4.3 Interfaces**

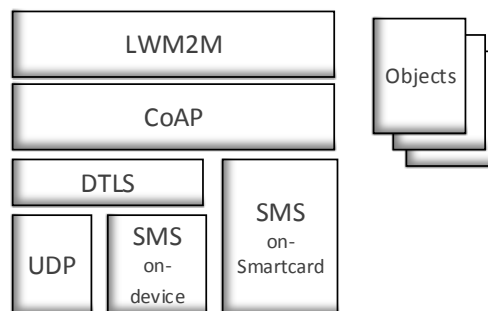
There are four interfaces supported by the reference point between LWM2M server and LWM2M client. The logical operation of each interface is defined as follows:

- 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).

## I.5 Protocols

### I.5.1 Protocol stack

The LWM2M has the protocol stack defined below. Figure I.5.1-1 shows the LWM2M protocol stack.

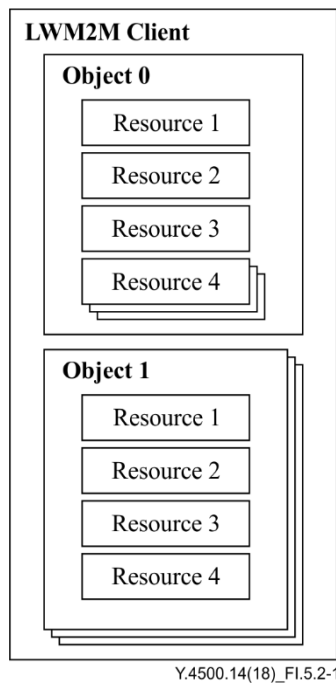


**Figure I.5.1-1 – LWM2M protocol stack**

- LWM2M objects: LWM2M objects are designed for the functionality provided by the LWM2M enabler. The LWM2M specification [b-OMA OMA-RD-LwM2M] 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 [b-IETF RFC 7252] 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 [b-IETF RFC 6347] is used to provide secure UDP/SMS on-device channels 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.

### I.5.2 Data model

In the LWM2M enabler technical specification [b-OMA OMA-RD-LwM2M], a simple data model is described. Basically, a resource made available by data model of the LWM2M client is a resource and resources are logically organized into objects. Figure I.5.2-1 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 I.5.2-1 – LWM2M data model [OMA TS LwM2M]**

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 be a single or multiple (possibility of several instantiations) one, 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.

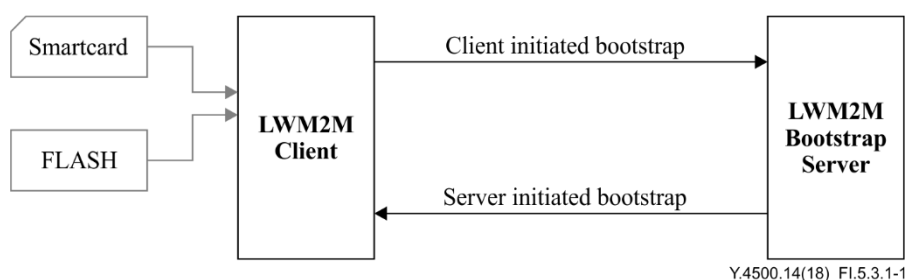
An object needs to be instantiated either by the LWM2M server or the LWM2M client, which is called object instance, before using the functionality of an object. After object instance is created, the LWM2M server can access that object instance and resources in the object instance. Furthermore a resource can contain a simple value (e.g., sensor measure), or a reference to an object instance.

### **I.5.3 Interface descriptions**

#### **I.5.3.1 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 (see Figure I.5.3.1-1) for bootstrapping:

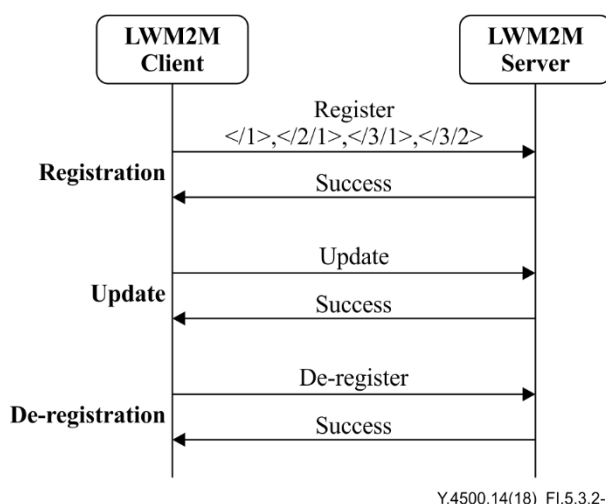
- Factory bootstrap: The LWM2M client is already provisioned at the time of the device manufacture. The pre-configured data is stored in the LWM2M client.
- Bootstrap from smartcard: When the device supports a smartcard and retrieval of bootstrap message from smartcard is successful, the LWM2M client processes the bootstrap message from the smartcard and applies it to the LWM2M client.
- Client initiated bootstrap: The LWM2M client requests and retrieves the bootstrap message from a LWM2M bootstrap server. In this case the LWM2M client needs to be pre-provisioned with the LWM2M bootstrap server bootstrap information.
- Server initiated bootstrap: The LWM2M bootstrap 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 server bootstrap information.



**Figure I.5.3.1-1 – Bootstrap modes**

### I.5.3.2 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 I.5.3.2-1 shows an example of registration procedure.

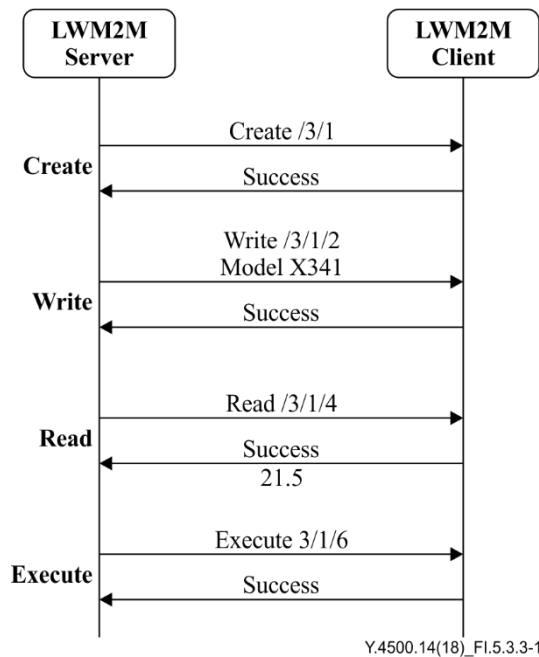


**Figure I.5.3.2-1 – Example of registration procedure**

### I.5.3.3 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.

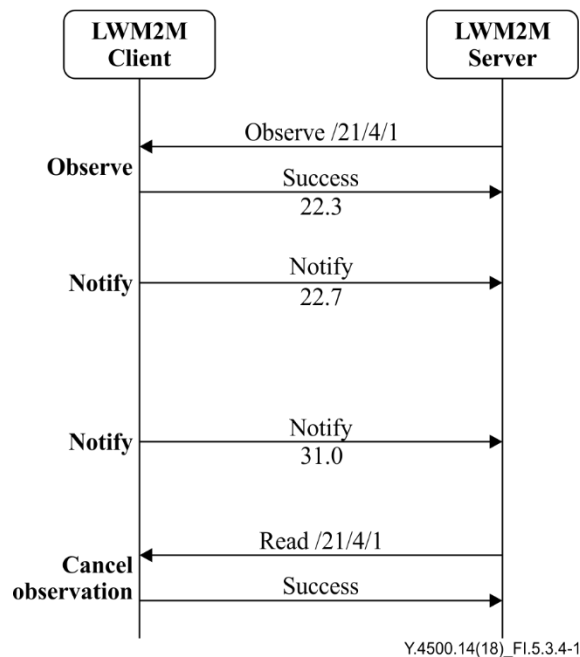
Figure I.5.3.3-1 shows an example of a device management and service enablement interface.



**Figure I.5.3.3-1 – Example of device management and service enablement interface**

#### **I.5.3.4 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. Figure I.5.3.4-1 shows an example of an information reporting interface.



**Figure I.5.3.4-1 – Example of information reporting interface**

## **I.6 Functions**

A first set of standard objects for the LWM2M 1.0 enabler have been developed:

- 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 rights 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: statistical information of network connection (e.g., SMS counter, UDP data size).

These standard objects are intended to support a variety of functionalities to manage LWM2M devices. OMA has already specified others LWM2M objects (e.g., as software management, device capability management,) and may create further objects in the future. Furthermore, other organizations and companies may define additional LWM2M objects for their own M2M services using according to the LWM2M object Template and Guideline Annex in [OMA TS LwM2M]: e.g., oneM2M has specified the set of LWM2M objects around CMDH policy functionality; IPSO Alliance has developed LWM2M objects for various sensors.

## Bibliography

- [b-IETF RFC 6347] IETF RFC 6347 (2012), *Datagram Transport Layer Security Version 1.2*. <https://tools.ietf.org/html/rfc6347>
- [b-IETF RFC 7252] IETF RFC 7252 (2014), *The Constrained Application Protocol (CoAP)*. <https://tools.ietf.org/html/rfc7252>
- [b-OMA OMA-RD-LwM2M] OMA OMA-RD-LightweightM2M-V1-0, *OMA Lightweight Machine to Machine Requirement*.
- [b-oneM2M.DR] oneM2M *Drafting Rules*.  
<http://www.onem2m.org/images/files/oneM2M-Drafting-Rules.pdf>.
- [b-oneM2M TS-0012] oneM2M Technical Specification TS-0012, *Base Ontology*.
- [b-ATIS.oneM2M.TS0014] ATIS.oneM2M.TS0014V200-2016  
<https://www.atis.org/docstore/search.aspx>
- [b-CCSA M2M-TS-0014] CCSA M2M-TS-0014-V2.0.0
- [b-ETSI TS 118 114] ETSI TS 118 114 V2.0.0 (2016), *oneM2M LWM2M Interworking* (oneM2M TS-0014 version 2.0.0 Release 2).  
[www.etsi.org/deliver/etsi\\_ts/118100\\_118199/118114/02.00.00\\_60/ts\\_118114v020000p.pdf](http://www.etsi.org/deliver/etsi_ts/118100_118199/118114/02.00.00_60/ts_118114v020000p.pdf)
- [b-TSDSI STD T1.oneM2M TS-0014] TSDSI STD T1.oneM2M TS-0014-2.0.0 V1.0.0  
<https://members.tdsi.in/index.php/s/2AgkUbCzaCKcDwE>
- [b-TTAT.MM-TS.0014] [b-TTAT.MM-TS.0014 v2.0.0]  
[http://www.tta.or.kr/data/ttas\\_view.jsp?rn=1&rn1=Y&rn2=&rn3=&nowpage=1&pk\\_num=TTAT.MM-TS.0014+v2.0.0&standard\\_no=TTAT.MM-TS.0014+v2.0.0&kor\\_standard=&publish\\_date=&section\\_code=&order=publish\\_date&by=desc&nowSu=1&totalSu=1&acode1=&acode2=&scode1=&scode2=](http://www.tta.or.kr/data/ttas_view.jsp?rn=1&rn1=Y&rn2=&rn3=&nowpage=1&pk_num=TTAT.MM-TS.0014+v2.0.0&standard_no=TTAT.MM-TS.0014+v2.0.0&kor_standard=&publish_date=&section_code=&order=publish_date&by=desc&nowSu=1&totalSu=1&acode1=&acode2=&scode1=&scode2=)
- [b-TTC TS-M2M-0014] TTC TS-M2M-0014v2.0.0  
[www.ttc.or.jp/jp/document\\_list/pdf/j/TS/TS-M2M-0014v2.0.0.pdf](http://www.ttc.or.jp/jp/document_list/pdf/j/TS/TS-M2M-0014v2.0.0.pdf)





## SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
<b>Series Y</b>	<b>Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities</b>
Series Z	Languages and general software aspects for telecommunication systems