|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **International Telecommunication Union** | | |
|  | |  | | |
| **ITU-T** | **Technical Report** | |
| TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU | | (15 September 2017) |
|  |  | | | |
|  | **YSTR-M2M-DG.AppDev**  **oneM2M – Application developer guide: Light control example using HTTP binding** | | | |
|  |  | | | |



|  |
| --- |
| Technical Report ITU-T YSTR-M2M-DG.AppDev  oneM2M – Application developer guide: Light control example using HTTP binding |

|  |
| --- |
| Summary  This Technical Report provides a simple use case for guiding application developers to develop applications using functionalities provided by a oneM2M service platform. |

|  |
| --- |
| History  This document contains Version 0 of the ITU-T Technical Report on "oneM2M-Application developer guide: Light control example using HTTP binding" approved at the ITU-T Study Group 20 meeting held in Geneva, 4-15 September 2017. |
|  |
| Keywords  Developer Guidance, HTTP, oneM2M. |

© 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**

Page

1 Scope 1

2 References 1

3 Terms and definitions 1

3.1 Terms defined elsewhere 1

3.2 Terms defined in this Technical Report 1

4 Abbreviations and acronyms 1

5 Conventions 2

6 Use case 2

7 Architecture 3

8 Procedures 4

8.1 Introduction 4

8.2 Call flows 4

8.3 Remote control scenarios 9

9 Implementation 10

9.1 Introduction 10

9.2 Assumptions 11

9.3 Addressing for entities 11

9.4 Modelling for light state data 12

9.5 Resource structure 12

9.6 Role of entities 13

9.7 Implementation procedures 14

Appendix A – Reading resources 39

A.1 Introduction 39

A.2 CSE resources 39

A.3 Gateway device application MN-AE 40

A.4 Light device applications 40

A.5 Smartphone application IN-AE 41

A.6 Access control policy 41

A.7 Containers 42

A.8 ContentInstances 43

A.9 Subscriptions 44

A.10 Groups 45

Technical Report ITU-T YSTR-M2M-DG.AppDev

oneM2M – Application developer guide:  
Light control example using HTTP binding

# 1 Scope

This Technical Report provides a guide for application developers to develop applications using functionalities provided by any oneM2M compliant service platform with the following scope:

– Objective of the use case;

– The architecture of the use case mapped into an oneM2M service platform;

– The execution procedures for implementation of the use case; and

– Implementation details of the use case.

# 2 References

[ITU-T Y.4500.1] Recommendation ITU-T Y.4500.1, *oneM2M –* *Functional architecture*.

[ITU-T Y.4500.4] Recommendation ITU-T Y.4500.4, *oneM2M – Service layer core protocol specification*.

[ITU-T Y.4500.9] Recommendation ITU-T Y.4500.9, *oneM2M – HTTP protocol binding*.

[ITU-T Y.4500.11] Recommendation ITU-T Y.4500.11, *oneM2M – Common terminology*.

# 3 Terms and definitions

## 3.1 Terms defined elsewhere

This Technical Report uses the terms and definitions given in [ITU‑T Y.4500.11].

## 3.2 Terms defined in this Technical Report

This Technical Report defines the following terms:

**3.2.1 M2M service provider domain**: Part of the M2M system that is associated with a specific M2M service provider.

**3.2.2 registrar CSE**: CSE (common services entity) where an application or another CSE has registered.

**3.2.3 resource**: Uniquely addressable entity in oneM2M architecture.

# 4 Abbreviations and acronyms

For the purposes of the Technical Report, the abbreviations given in [ITU‑T Y.4500.11] and the following apply:

ACP Access Control Policy

ADN Application Dedicated Node

ADN-AE AE which resides in the ADN

AE Application Entity

CoAP Constrained Application Protocol

CSE Common Services Entity

CSE-ID CSE Identifier

DNS Domain Name System

FQDN Fully Qualified Domain Name

HTTP Hypertext Transfer Protocol

IN Infrastructure Node

IN-AE AE that is registered with the CSE in the IN

IN-CSE CSE which resides in the IN

JSON JavaScript Object Notation

M2M Machine to Machine

Mca Reference Point for M2M Communication with AE

Mcc Reference Point for M2M Communication with CSE

MN Middle Node

MN-AE AE that is registered with the CSE in MN

MN-CSE CSE which resides in the MN

PoA Point of Access

SP Service Provider

URI Uniform Resource Identifier

XML extensible Markup Language

# 5 Conventions

None.

# 6 Use case

This guide is based on a home lighting use case involving lights in a home that can be remotely controlled by a user's smartphone leveraging the capabilities of oneM2M. An overview of the use case is shown in Figure 6-1. The main components are introduced as follows:

1) The lights are deployed in a home and are attached to a home gateway.

2) The home gateway communicates with a cloud service platform allowing the lights to be controlled remotely by the smartphone.

3) The cloud service platform supports a set of services to enable the smartphone to more easily control the lights in the home. Some examples of services include registration, discovery, data management, group management, subscription/notification.

4) The smartphone hosts an application used to remotely control the lights in the home and supports the following capabilities:

– discovery of lights deployed in the home;

– sending commands to change light states i.e., ON and OFF;

– retrieval of light states.

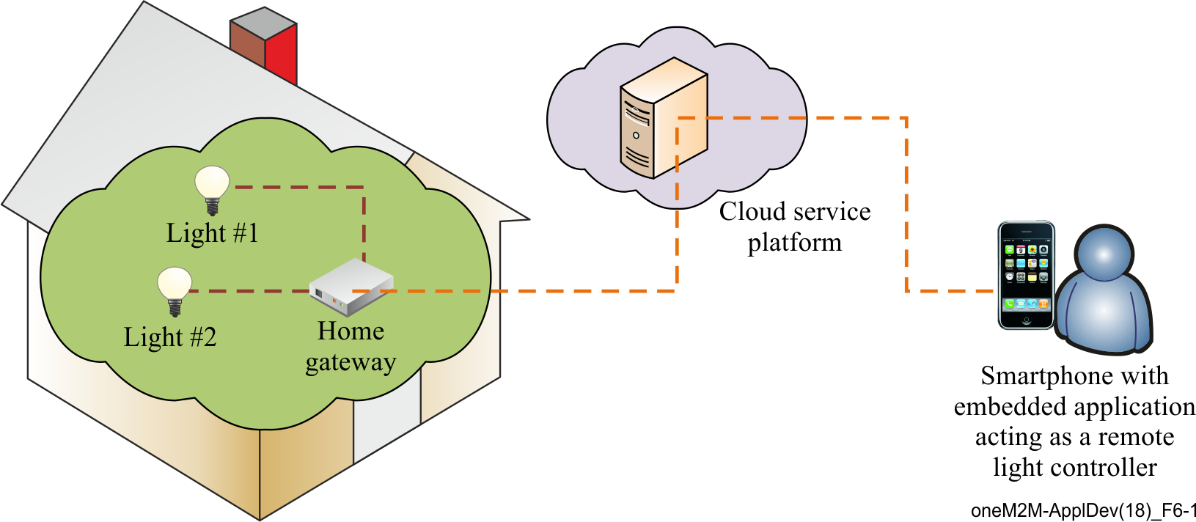


Figure 6-1 – Overview of remote lights control use case

# 7 Architecture

This clause describes how the different components of this use case can be represented by corresponding oneM2M architectural entities as shown in Figure 7-1.

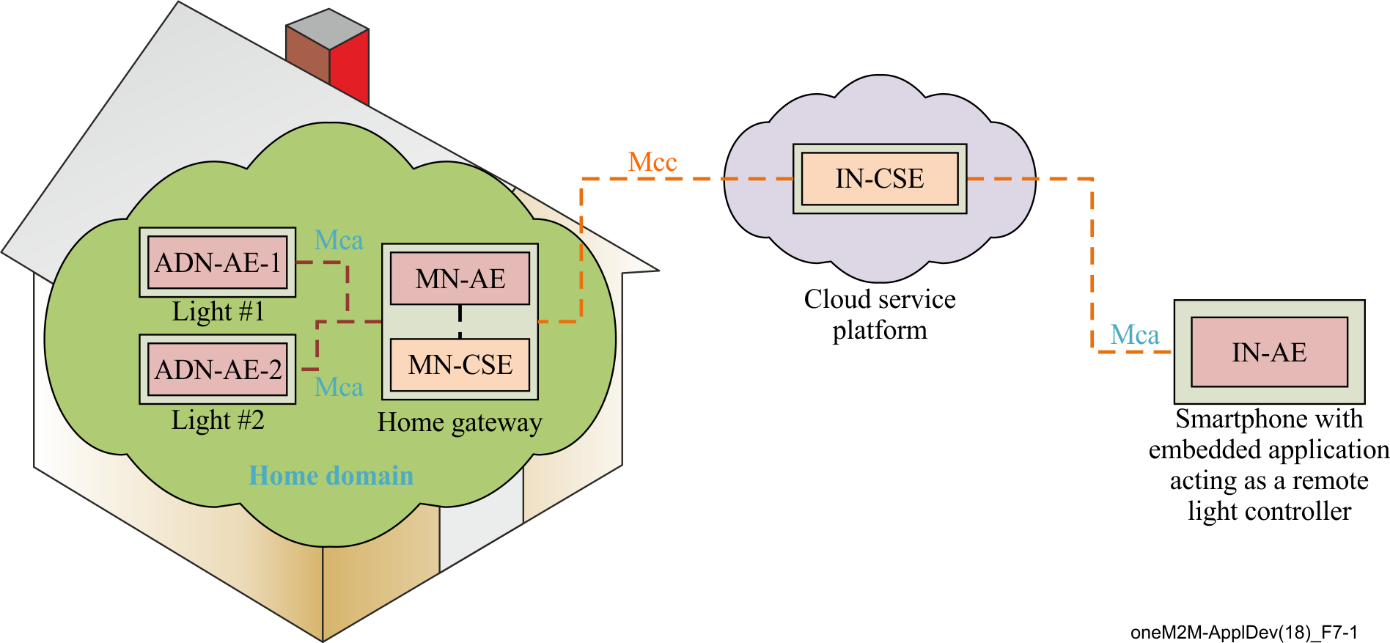


Figure 7-1 – oneM2M functional architecture of remote lights control use case

In the oneM2M functional architecture two basic types of entities are defined. One is an application entity (AE) and the other is a common services entity (CSE). In this use case, the lights and smartphone each host an AE. Also, an infrastructure node (IN)-CSE is hosted in the cloud by the oneM2M service provider (SP) and a middle node (MN)-CSE is hosted on the home gateway.

The oneM2M defined Mca reference point is used to interface an AE and CSE. The oneM2M defined Mcc reference point is used to interface CSEs. In this use case, the reference point used between a light AE and home gateway MN-CSE or smartphone AE and IN-CSE is Mca while the reference point used between the home gateway MN-CSE and the oneM2M service platform IN-CSE is Mcc.

In summary, applications used in the current use case are classified as follows:

1) Application dedicated node (ADN)-AE1: an application embedded in *Light#1* with capabilities to control *Light#1* and interact with the home gateway MN-CSE through *Mca* reference point;

2) ADN-AE2: an application embedded in *Light#2* with capabilities to control *Light#2* and interact with the home gateway MN-CSE through *Mca* reference point;

3) IN-AE: a smartphone application embedded in the smartphone device with capabilities to interact directly with the oneM2M service platform IN-CSE through *Mcc* reference point and thereby remotely control *Light#1* and *Light#2*;

4) MN-AE: a gateway application embedded into the home gateway that interacts with the MN‑CSE through *Mca* reference point.

# 8 Procedures

## 8.1 Introduction

The deployment of the oneM2M standard in the present use case requires procedures that are classified as follows:

1) **Registration**: the current procedure contains light application registration, gateway application registration, and accessControlPolicy resource creation for selective access to data storage resources.

2) **Initial resource creation**: the current procedure contains group resource creation, container resources creation with specific access control policies, content instance resources creation with initial light states, subscription resources creation for notifications.

3) **Discovery of container resource**: all containers with a specific filter criteria are discovered by the gateway application and then configured as members of a group resource.

4) **Discovery and retrieval lights states**: all containers with a specific filter criteria are discovered and retrieved using resource identities through a smartphone application which gains access to the oneM2M service platform so content information can be retrieved.

5) **Single light switch on/off**: any light that is discovered by and connected to the smartphone application is able to be switched on and off via a smartphone application.

6) **Multiple lights switch on/off**:multiple lights that are discovered are able to be switched on and off together via a smartphone application.

## 8.2 Call flows

### 8.2.1 Application registration and access control policy creation

Call flows regarding the registration phase depicted in Figure 8.2.1-1 are ordered as follows:

1) Gateway (MN-CSE) registers with the oneM2M service platform (IN-CSE).

2) Gateway application (MN-AE) registers with the gateway (MN-CSE).

3) Light applications (ADN-AE1 and ADN-AE2) register with the gateway (MN-CSE).

4) Smartphone application (IN-AE) registers with the oneM2M service platform (IN-CSE).

5) Gateway application (MN-AE) discovers the smartphone application (IN-AE) from gateway (MN-CSE) with specific filter criteria. The discovered IN-AE is granted access to the remote light control service containers.

6) Gateway application (MN-AE) creates an accessControlPolicy resource granting all the entities playing roles in the current use case including ADN-AE1, ADN-AE2, MN-AE and IN-AE access to the created container and content instance resources.

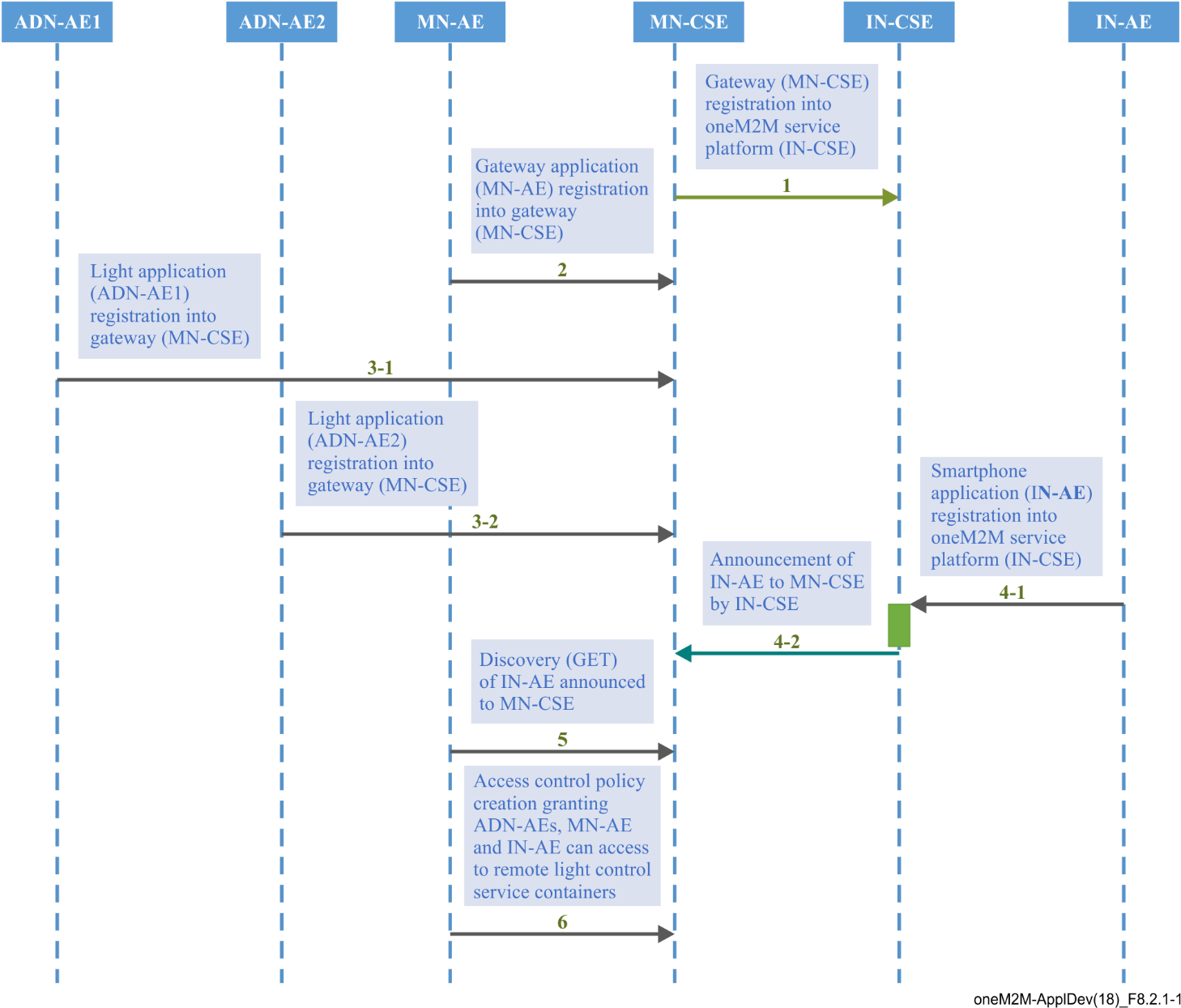


Figure 8.2.1-1 – Registration phase call flows

### 8.2.2 Initial resource creation

Call flows regarding the initial resource creation phase depicted in Figure 8.2.2-1 are ordered as follows:

1) Gateway application (MN-AE) creates a group resource on gateway (MN-CSE), for updating and retrieving group light state named as *containers\_group*. The group members are added from the list of discovered container resources that the MN-AE discovers. The group resource is created with a link to the same access control policy (ACP).

2) Two container resources are created in the gateway (MN-CSE) to store the light states under the registered light application ADN-AE1 and ADN-AE2, respectively. The containers are created with a link to the same ACP.

3) Content instance resources are created by light applications (ADN-AE1 and ADN-AE2) under each created container and represent the controlled light states.

4) Subscription resources are created under the containers in the gateway (MN-CSE) so that subscribers, i.e., light applications, can be notified whenever there is a new contentInstance resource created by the IN-AE.

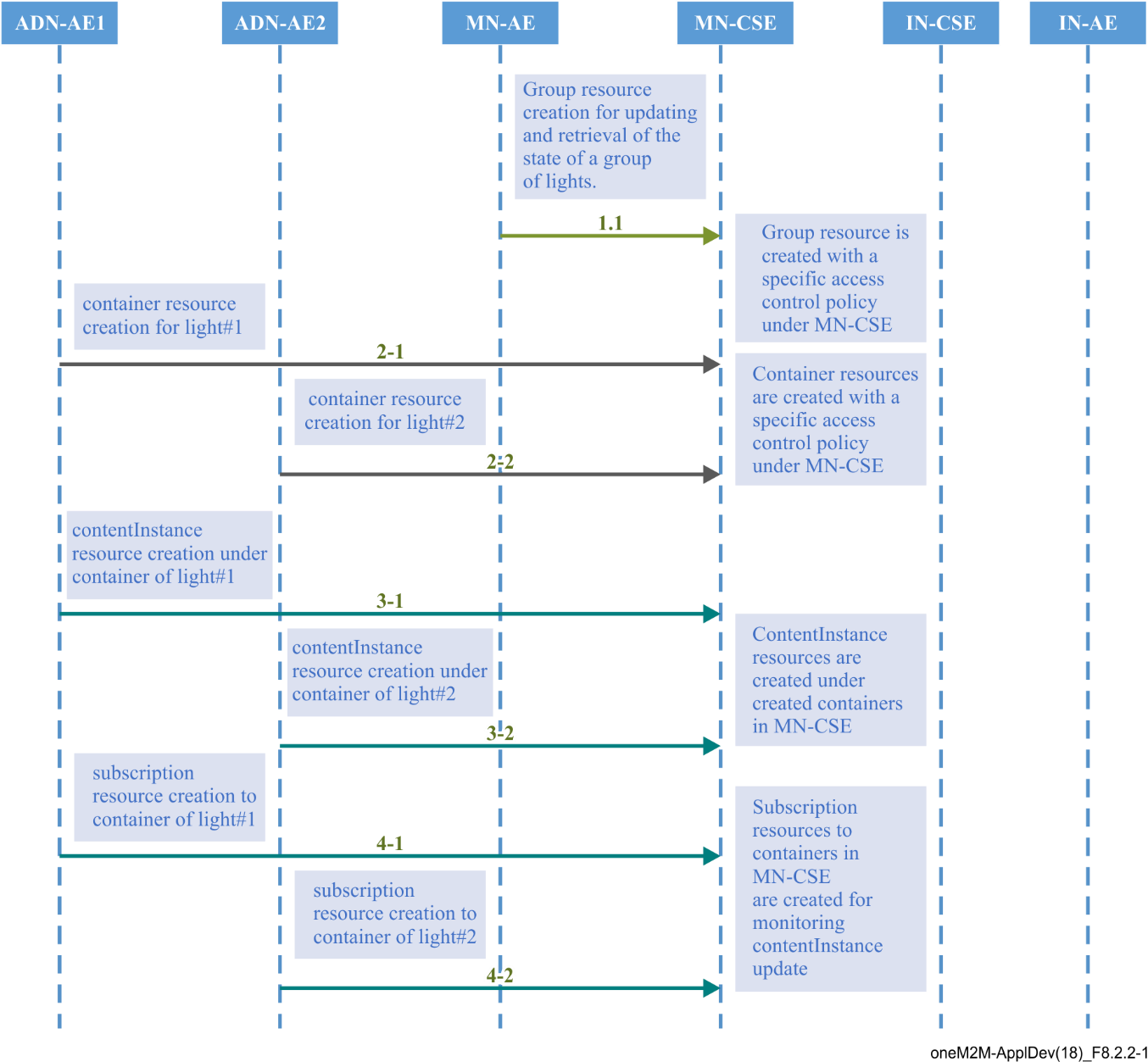


Figure 8.2.2-1 – Initial resource creation phase call flows

### 8.2.3 Discovery of group resources

Call flows regarding the discovery and update of group resources are depicted in Figure 8.2.3-1 and ordered as follows:

1) Gateway application (MN-AE) periodically sends a RETRIEVE request including the parameter *filterUsage* and specific filter criteria condition(s) as a query string for discovery of container resources stored in the MN-CSE of gateway. The filter criteria conditions for the discovery operation include *createdBefore, createdAfter, modifiedSince, unmodifiedSInce, label, creator, expireAfter, resourceType,* etc.

2) Gateway (MN-CSE) responds with uniform resource identifiers (URIs) of the discovered container resources, if any, to the gateway application (MN-AE) according to the filter criteria(s).

3) Gateway application (MN-AE) sends an update request to update the list of group members within the previously created group resource with identifiers of the discovered containers.

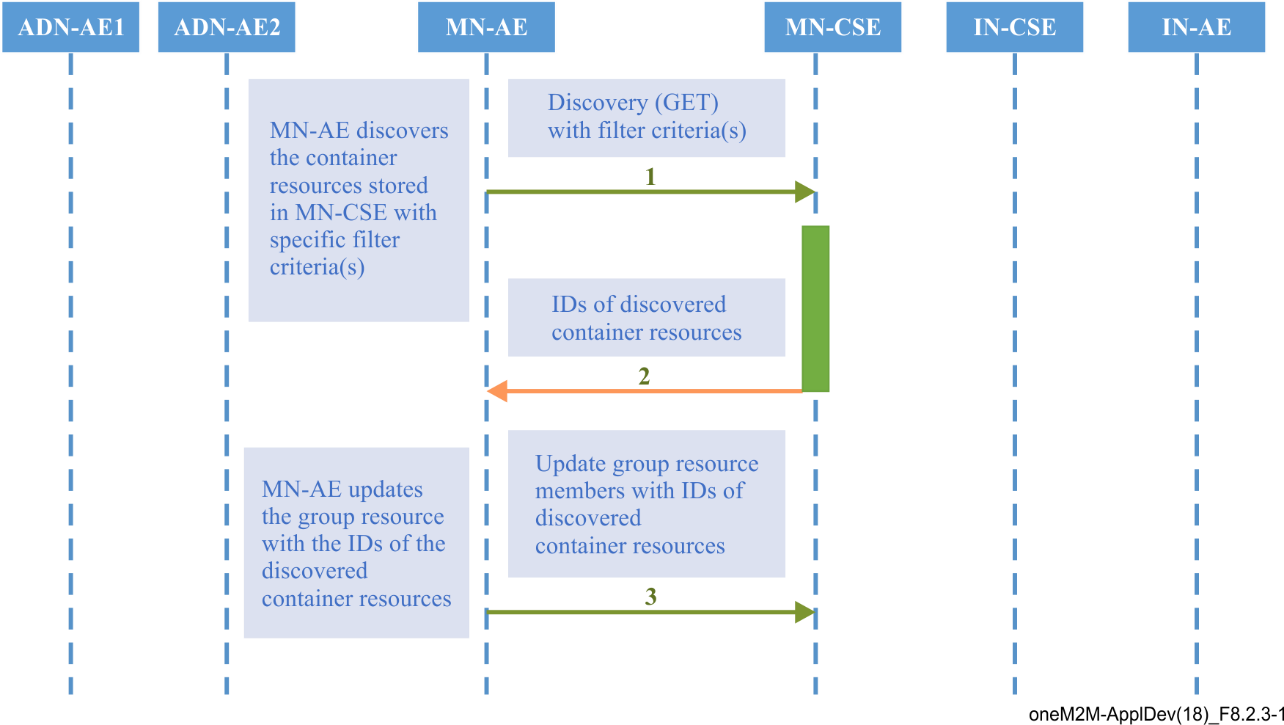


Figure 8.2.3-1 – Discovery and group light state update phase call flows

### 8.2.4 Discovery and retrieval of contentInstance resources

Call flows regarding the discovery and retrieval of contentInstance resources depicted in Figures 8.2.4‑1 and 8.2.4-2 are ordered as follows:

1) The smartphone application (IN-AE) periodically sends a RETRIEVE request including the parameter *filterUsage* and specific filter criteria condition(s) as a query string for discovery of container resources stored in the MN-CSE of gateway.

The IN-AE also sends a Discovery request to the MN-CSE for the discovery of the group resources located in the MN-CSE.

2) The gateway (MN-CSE) responds to the IN-AE with URIs of the discovered container resources under ADN-AE1 and ADN-AE2, if any.

For the case where the IN-AE sends a Discovery request for the discovery of group resources, the MN-CSE responds to the IN-AE with the URIs of the discovered group resources located in the MN-CSE, if any.

3) The IN-AE sends GET requests for retrieval of the latest contentInstance resources from each discovered light container resource.

In the case of retrieval of the latest contentInstance resources of the group of containers, the IN-AE sends a RETRIEVE request to the *fanOutPoint* of the discovered group resource.

4) The MN-CSE responds to the IN-AE with the latest light state(s).

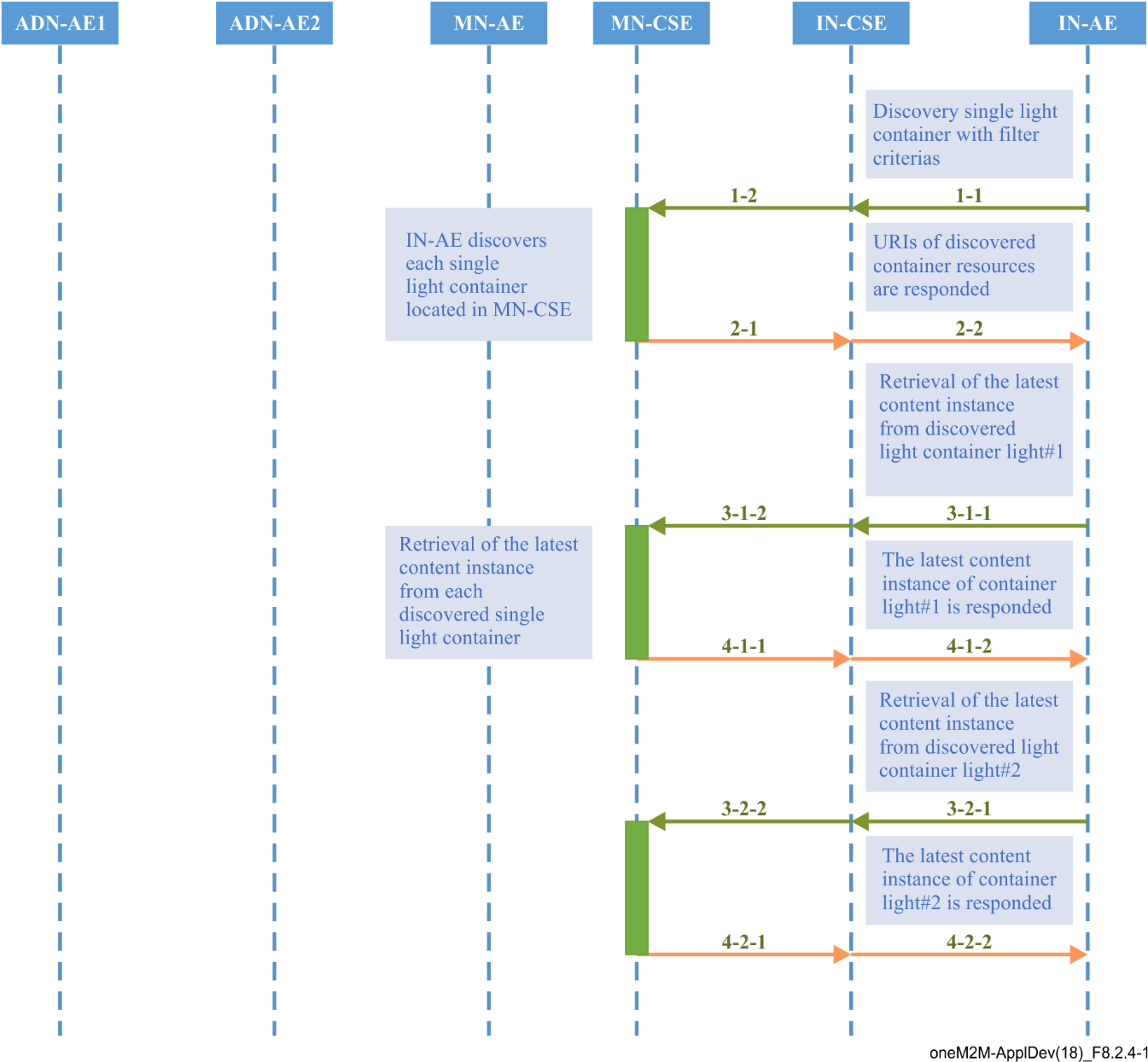


Figure 8.2.4-1 – Discovery and single light retrieval phase call flows

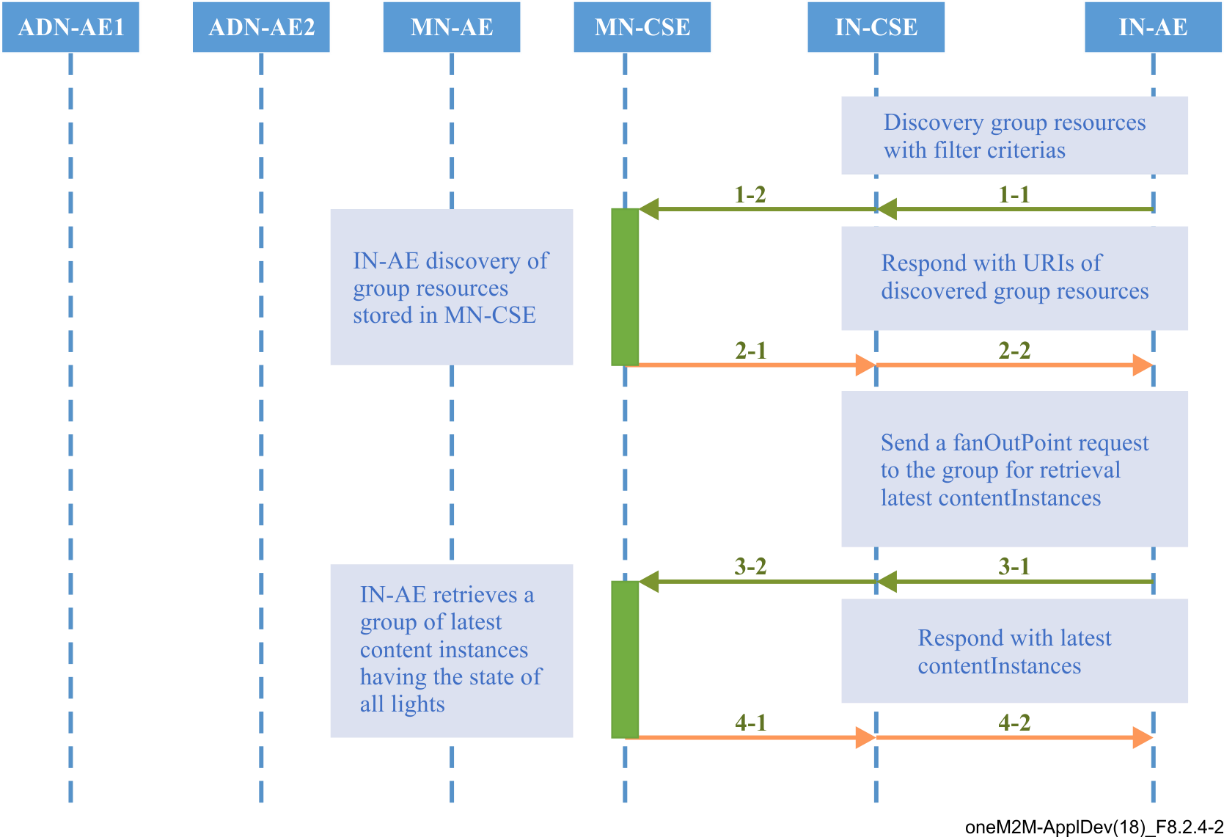


Figure 8.2.4-2 – Discovery and a group of lights retrieval phase call flows

## 8.3 Remote control scenarios

### 8.3.1 Introduction

Lights are able to be controlled remotely through the smartphone application accessing the oneM2M service platform. Two scenarios are introduced in clauses 8.3.2 and 8.3.3.

### 8.3.2 Single light control

*Light#1* and *Light#2* are controlled remotely by a human user through a smartphone application (IN‑AE). A call flow for single light control is depicted in Figure 8.3.2-1 and the steps are ordered as follows:

1) When the user updates the light state on their smartphone, the IN-AE creates a new contentInstance representing a new light state under the targeted container of a Light ADN‑AE stored in the MN-CSE.

2) If the contentInstance is created successfully*,* the MN-CSE sends a notification to the corresponding Light ADN-AE to notify it that a new contentInstance resource was created.

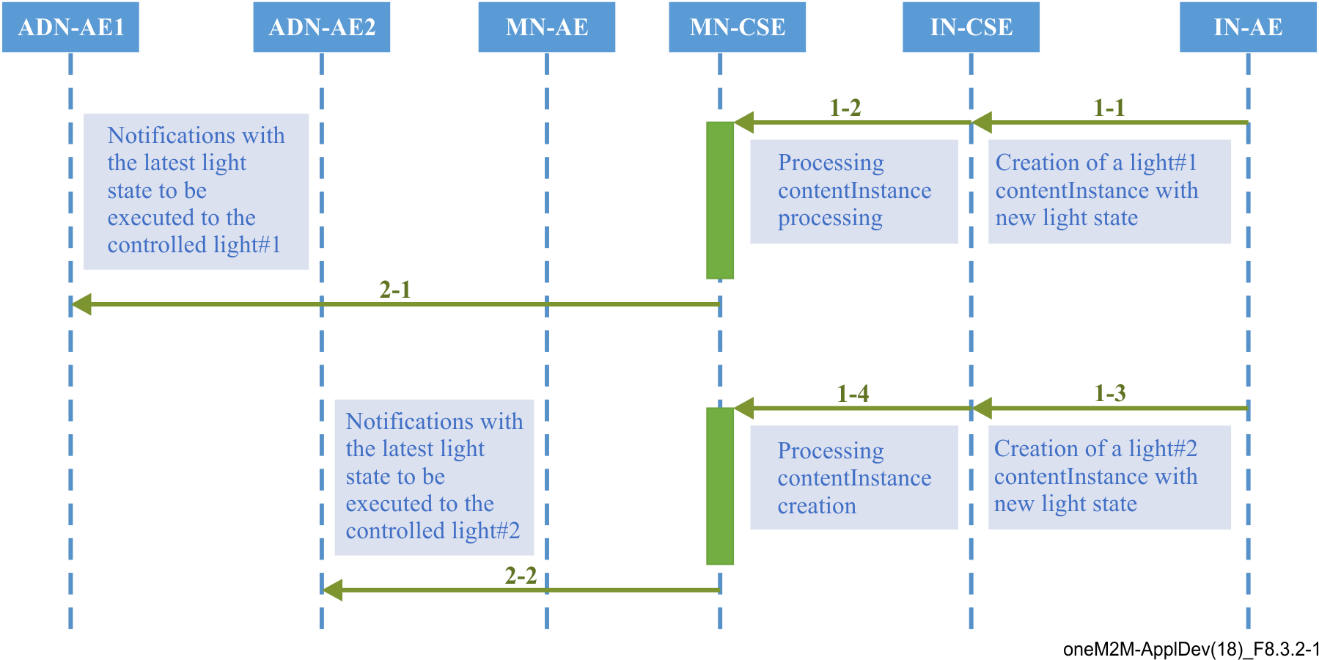


Figure 8.3.2-1 – Single light remote control phase call flows

### 8.3.3 Multiple light control

Users can also remotely control multiple lights through the smartphone application (IN-AE) by sending a single light control command to the group resource. A call flow for multiple lights control is depicted in Figure 8.3.3-1 and the steps are ordered as follows:

1) When the user updates the state of a group of lights on their smartphone, the IN-AE sends a contentInstance create request targeting the group resource on the MN-CSE. The MN-CSE then fans out the request to the individual Light container member resources on the MN-CSE.

2) For each contentInstances created successfully*,* the MN-CSE sends a notification to the corresponding Light ADN-AE.

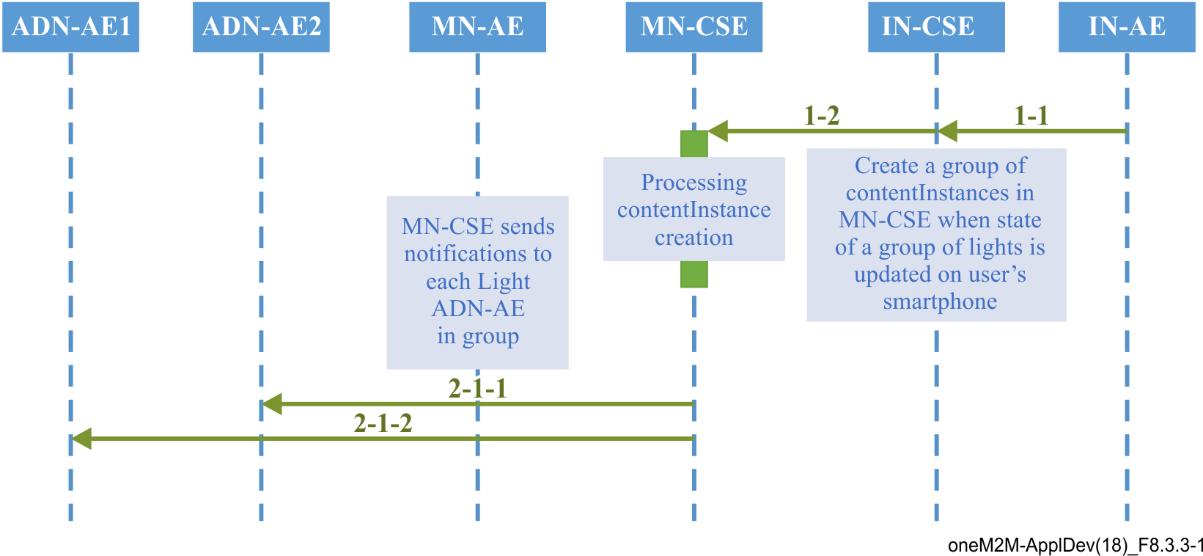


Figure 8.3.3-1 – Multiple lights remote control phase call flows

# 9 Implementation

## 9.1 Introduction

Clause 9 presents necessary procedures required for the implementation of the remote lights control use case, including conditions that must be met for the correct implementation of the current use case, and resource tree, etc.

## 9.2 Assumptions

Assumptions are presented below in order to ensure the remote lights control use case can be correctly implemented.

– all applications are server capable;

– devices and application entities are independently addressable with host names resolved by domain name system (DNS) network services;

– host port number 8080 is reserved for oneM2M services;

– security is not considered in the current use case;

– hypertext transfer protocol (HTTP) binding of oneM2M primitives is used in the current use case;

– both extensible markup language (XML) and JavaScript object notation (JSON) serializations of oneM2M primitives are used in the current use case;

– all mandatory HTTP headers are presented in the HTTP requests while optional headers are selectively used in the current use case;

– all mandatory resource attributes for resources presented in the current use case are presented in the HTTP requests while optional resource attributes are selectively used in the current use case;

– the IN-CSE and MN-CSE in the current use case are deployed within the same oneM2M SP domain;

– all AEs in the current use case are initially registered with CSEs and the identifier of the AEs are assigned by the Registrar CSE of the AE accordingly, starting with a character of 'C';

– all resources created in the current use case are addressable with the oneM2M resource identifier form of *hierarchical address*;

– short names for the representation of the resources and attributes are used in the current use case;

– default ACP has already been created under IN-CSE and it is used for MN‑CSE registration with IN-CSE;

– all request originators send *Blocking Requests* for accessing resources located in CSEs.

## 9.3 Addressing for entities

Each oneM2M entity, including AE and CSE, is addressable with a correct host address that can be IP addresses or fully qualified domain name (FQDN) addresses resolved to IP addresses by DNS network services according to addressing rules specified in oneM2M standards.

The IN-CSE and MN-CSE entities presented in this use case are addressable with the following identifiers:

– IN-CSE:

• CSE-ID: in-cse

• resourceName of IN-CSE's CSEBase resource: server

– MN-CSE:

• CSE-ID: mn-cse

• resourceName of MN-CSE's CSEBase resource: home\_gateway

## 9.4 Modelling for light state data

The light state *ON* or *OFF* stored as the content of content instance resource is modelled as string in XML representation and can be represented as *<con>ON</con>* or *<con>OFF</con>,* while represented as *{ … "con":"ON" …}* or *{… "con":"OFF" …}*, respectively.

## 9.5 Resource structure

### 9.5.1 Introduction

The development of an oneM2M application includes the design of the resource trees of service capability layers i.e., IN-CSE and MN-CSE in the current use case. The resource tree is constructed with child resources created according to the high-level procedures presented in the oneM2M application developer guide clause 7. All child resources shown in the resource trees are mandatorily required in order to correctly implement the remote lights control use case.

### 9.5.2 Resource structure of IN-CSE

The resource tree of IN-CSE starts with a CSEBasenamed ***server*** depicted in Figure 9.5.1-1.

The root CSEBase has two direct child resources, a remoteCSEnamed *home\_gateway* and an AEnamed *smartphone\_ae*.

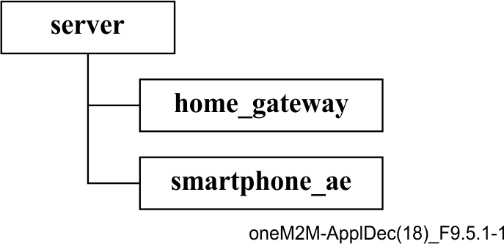


Figure 9.5.1-1 – IN-CSE resource structure

### 9.5.3 Resource structure of MN-CSE

The resource tree of MN-CSE starts with a CSEBasenamed ***home\_gateway*** depicted in Figure 9.5.2‑1.

The resource tree of MN-CSE is constructed with child resources as follows:

– an accessControlPolicynamed *gateway\_acp*;

– an ADN-AEnamed *light\_ae1* which contains sub-resources of a containernamed *light* and multiplecontentInstances;

– an ADN-AEnamed *light\_ae2* which contains sub-resources of a containernamed *light* and multiplecontentInstances;

– light\_state\_sub1 and light\_state\_sub2 subscriptionresources that are child resources of the two *light* containers;

– a MN-AEnamed *gateway\_ae* which contains a group child resource;

– a group resource named *containers\_group* whose members are the *light* containers of each AND-AEs. Smartphone application users with proper access control privileges can send a *fanOutPoint* request to this group to create and retrieve content instances in the two *light* containers.

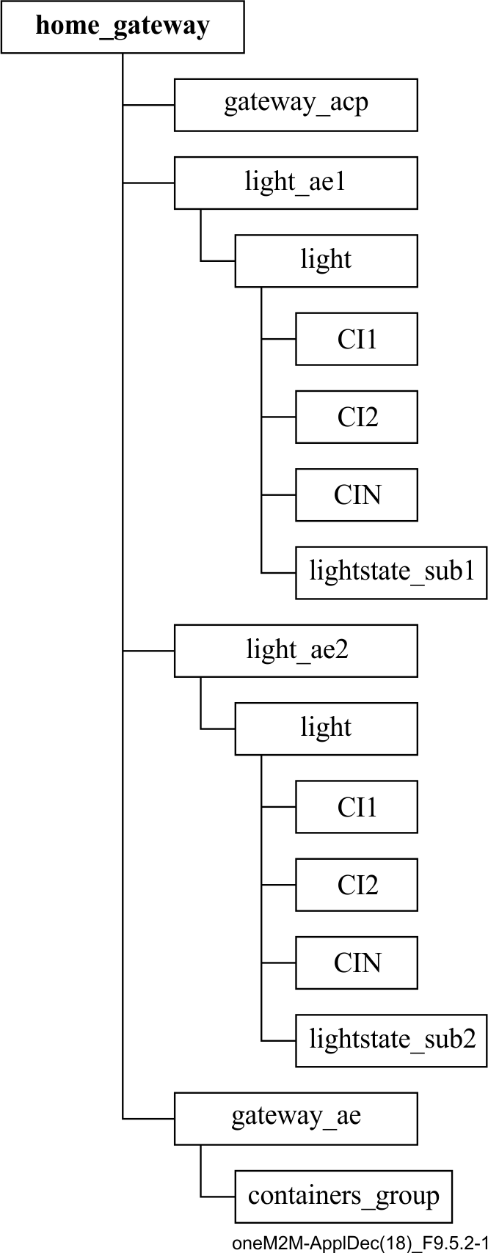


Figure 9.5.2-1 – MN-CSE resource tree

## 9.6 Role of entities

### 9.6.1 oneM2M service platform (IN-CSE)

The oneM2M service platform is modelled as an IN-CSE and is responsible for:

– handling the registration requests fromthesmartphone AE and home gateway MN-CSE.

### 9.6.2 Home gateway application (MN-AE)

The home gateway application is modelled as a MN-AE and is responsible for:

– initializing the home gateway device;

– creating an ACP resource gateway\_acpin the MN-CSE;

– registering the home gateway application with the MN-CSE;

– creating the group resource with ACP gateway\_acp in the MN-CSE;

– discovering device applications registered with the MN-CSE.

### 9.6.3 Light applications (ADN-AE1 and ADN-AE2)

Each of the light applications are modelled as an ADN-AE and are responsible for:

– initializing the light control device;

– registering the light devices with the MN-CSE;

– creating container resources named "*light*" with ACP gateway\_acp in the MN-CSE, respectively;

– creating subscription resources lightstate\_sub1 and lightstate\_sub2 under the two *light* containers; and

– creating content instance resources under containers light1 and light2 with initial light state, respectively.

### 9.6.4 Smartphone application (IN-AE)

The smartphone application is modelled as an IN-AE, which directly communicates with the oneM2M service platform IN-CSE and is responsible for:

– initializing the smartphone light control application;

– registering the smartphone application with the IN-CSE;

– discovering the two *light* containers;

– displaying the discovered light states;

– accepting the light state modification commands from the smartphone application user;

– executing the light state modification commands for single and multiple lights.

## 9.7 Implementation procedures

### 9.7.1 Introduction

The implementation procedures in the current use case are mapped into HTTP bindings with both XML and JSON serializations of oneM2M primitives according to the standard APIs describing the reference points Mca and Mcc, as defined in [ITU-T Y. 4500.1], [ITU-T Y.4500.4] and the HTTP binding [ITU-T Y.4500.9].

In addition, *short names* for the representation of the resources and attributes are used in the implementation procedures.

### 9.7.2 MN-CSE registration

The implementation starts with the registration of MN-CSE with IN-CSE as shown in the following procedure.

The following example shows the MN-CSE registration request and response using XML serialization:

HTTP Request:

POST /~/in-cse/server?rcn=0 HTTP/1.1

Host: in.provider.com:8080

X-M2M-Origin: /mn-cse

Content-Type: application/xml;ty=16

X-M2M-RI: incse-88643

<?xml version="1.0" encoding="UTF-8"?>

<m2m:csr xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="home\_gateway">

<csi>mn-cse</csi>

<cb>mn.provider.com/mn-cse</cb>

<rr>true</rr>

<poa>http://mn.provider.com:8080</poa>

<cst>2</cst>

<acpi>/in-cse/acp-666957710</acpi>

</m2m:csr>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: incse-88643

Content-Location: /in-cse/csr-299409504

The following example shows the MN-CSE registration request and response using HTTP with JSON serialization:

HTTP Request:

POST /~/in-cse/server?rcn=0 HTTP/1.1

Host: in.provider.com:8080

X-M2M-Origin: /mn-cse

Content-Type: application/json;ty=16

X-M2M-RI: incse-88643

{

"m2m:csr":

{

"rn": "home\_gateway",

"csi": "mn-cse",

"cb": "mn.provider.com/mn-cse",

"rr": true,

"poa": ["http://mn.provider.com:8080"],

"cst": 2,

"acpi": ["/in-cse/acp-666957710"]

}

}

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: incse-88643

Content-Location: /in-cse/csr-299409504

### 9.7.3 Access control policy creation

When an ACP resource is created, a list of one or more allowed request originators must be specified in the *acor* field and the allowed operations in the *acop* field.

The value of *acop* is set to 63, which indicates that the specified originator is granted privileges to conduct *CREATE, RETRIEVE, UPDATE, DELETE, DISCOVERY,* and *NOTIFY* operations.

The creation of ACP resource *gateway\_acp* in MN-CSE is implemented in the following procedure.

The following example shows an ACP create request and response using HTTP with XML serialization.

HTTP Request:

POST /~/mn-cse/home\_gateway?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: Cgateway\_ae

Content-Type: application/xml;ty=1

X-M2M-RI: mncse-62948

<m2m:acp xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="gateway\_acp">

<pv>

<acr>

<acor>Cgateway\_ae Clight\_ae1 Clight\_ae2 /in-cse/Csmartphone\_ae</acor>

<acop>63</acop>

</acr>

</pv>

<pvs>

<acr>

<acor>Cgateway\_ae </acor>

<acop>51</acop>

</acr>

</pvs>

</m2m:acp>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-62948

Content-Location: /mn-cse/acp-805496226

The following example shows an ACP create request and response using HTTP with XML serialization:

HTTP Request:

POST /~/mn-cse/home\_gateway?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: Cgateway\_ae

Content-Type: application/json;ty=1

X-M2M-RI: mncse-62948

{

"m2m:acp":

{

"rn": "gateway\_acp",

"pv":

{

"acr":[{

"acor":

[

"Cgateway\_ae",

"Clight\_ae1",

"Clight\_ae2",

"/in-cse/Csmartphone\_ae"

],

"acop":63

}]

},

"pvs":

{

"acr":[{

"acor":

[

"Cgateway\_ae"

],

"acop":51

}]

}

}

}

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-62948

Content-Location: /mn-cse/acp-805496226

The ACP resource is used to grant applications the access rights to conduct specific operations and access to specific resources. The list of applications could be obtained with a discovery procedure using filter criteria conditions. For more details about the discovery procedure, see clause 9.7.9. Here it is required that the list of applications has been discovered before creating the ACP resource.

### 9.7.4 Application entities registration

#### 9.7.4.1 Light application ADN-AE1

The registration of ADN-AE1 with MN-CSE is shown in the following procedure. The ACP identifier (unstructured SP-relative resourceID) which is assigned to ADN-AE1 is /mn-cse/acp-805496226.

The following example shows an ADN-AE registration request and response using HTTP with XML serialization:

HTTP Request:

POST /home\_gateway?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: C

Content-Type: application/xml;ty=2

X-M2M-RI: mncse-92345

<?xml version="1.0" encoding="UTF-8"?>

<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="light\_ae1">

<api>A01.com.company.lightApp1</api>

<rr>true</rr>

<poa>http://192.168.0.10:9090</poa>

<acpi>/mn-cse/acp-805496226</acpi>

</m2m:ae>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-92345

Content-Location: /mn-cse/ae-CAE340304071

The following example shows an ADN-AE registration request and response using HTTP with JSON serialization:

HTTP Request:

POST /home\_gateway?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: C

Content-Type: application/json;ty=2

X-M2M-RI: mncse-92345

{

"m2m:ae":

{

"rn": "light\_ae1",

"api": "A01.com.company.lightApp1",

"rr": true,

"poa": ["http://192.168.0.10:9090"],

"acpi": ["/mn-cse/acp-805496226"]

}

}

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-92345

Content-Location: /mn-cse/ae-CAE340304071

#### 9.7.4.2 Light application ADN-AE2

The registration of ADN-AE2 with MN-CSE is shown in the following procedure. The ACP identifier (unstructured SP-relative resourceID) which is assigned to ADN-AE2 is /mn-cse/acp-805496226.

The following example shows an ADN-AE registration request and response using HTTP with XML serialization.

HTTP Request:

POST /home\_gateway?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: C

Content-Type: application/xml;ty=2

X-M2M-RI: mncse-18346

<?xml version="1.0" encoding="UTF-8"?>

<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="light\_ae2">

<api>A01.com.company.lightApp2</api>

<rr>true</rr>

<poa>http://192.168.0.20:9090</poa>

<acpi>/mn-cse/acp-805496226</acpi>

</m2m:ae>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-18346

Content-Location: /mn-cse/ae-CAE340304042

The following example shows an ADN-AE registration request and response using HTTP with JSON serialization:

HTTP Request:

POST /home\_gateway?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: C

Content-Type: application/json;ty=2

X-M2M-RI: mncse-18346

{

"m2m:ae":

{

"rn": "light\_ae2",

"api": "A01.com.company.lightApp2",

"rr": true,

"poa": ["http://192.168.0.20:9090"],

"acpi": ["/mn-cse/acp-805496226"]

}

}

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-18346

Content-Location: /mn-cse/ae-CAE340304042

#### 9.7.4.3 Home gateway application MN-AE

The registration of MN-AE with MN-CSE is shown in the following procedure. The ACP identifier (unstructured SP-relative resourceID) which is assigned to MN-AE is /mn-cse/acp-805496226.

The following example shows an MN-AE registration request and response using HTTP with XML serialization:

HTTP Request:

POST /home\_gateway?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: C

Content-Type: application/xml;ty=2

X-M2M-RI: mncse-19347

<?xml version="1.0" encoding="UTF-8"?>

<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="gateway\_ae">

<api>A01.com.company.gatewayApp</api>

<rr>false</rr>

<acpi>/mn-cse/acp-805496226</acpi>

</m2m:ae>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-19347

Content-Location: /mn-cse/ae-CAE340303271

The following example shows an MN-AE registration request and response using HTTP with JSON serialization:

HTTP Request:

POST /home\_gateway?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: C

Content-Type: application/json;ty=2

X-M2M-RI: mncse-19347

{

"m2m:ae":

{

"rn": "gateway\_ae",

"api": "A01.com.company.gatewayApp",

"rr": false,

"acpi": ["/mn-cse/acp-805496226"]

}

}

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-19347

Content-Location: /mn-cse/ae-CAE340303271

#### 9.7.4.4 Smartphone application IN-AE

The registration of IN-AE with IN-CSE is shown in the following procedure. The ACP identifier (unstructured SP-relative resourceID) which is assigned to IN-AE is /in-cse/acp-666957710.

The following example shows an IN-AE registration request and response using HTTP with XML serialization:

HTTP Request:

POST /server?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: C

Content-Type: application/xml;ty=2

X-M2M-RI: incse-16346

<?xml version="1.0" encoding="UTF-8"?>

<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="smartphone\_ae">

<api>A01.com.company.lightControlApp</api>

<rr>false</rr>

<acpi>/in-cse/acp-666957710</acpi>

</m2m:ae>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: incse-16346

Content-Location: /in-cse/ae-CAE340304178

The following example shows an IN-AE registration request and response using HTTP with JSON serialization:

HTTP Request:

POST /server?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: C

Content-Type: application/json;ty=2

X-M2M-RI: incse-16346

{

"m2m:ae":

{

"rn": "smartphone\_ae",

"api": "A01.com.company.lightControlApp",

"rr": false,

"acpi": ["/in-cse/acp-666957710"]

}

}

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: incse-16346

Content-Location: /in-cse/ae-CAE340304178

### 9.7.5 Containers creation

#### 9.7.5.1 Create a container of ADN-AE1

The creation of a container resource for ADN-AE1 is shown in the following procedure.

The following example shows a container create request and response using HTTP with XML serialization:

HTTP Request:

POST /home\_gateway/light\_ae1?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: Clight\_ae1

Content-Type: application/xml;ty=3

X-M2M-RI: mncse/13345

<m2m:cnt xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="light">

</m2m:cnt>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-13345

Content-Location: /mn-cse/cnt-582759912

The following example shows a container create request and response using HTTP with JSON serialization:

HTTP Request:

POST /home\_gateway/light\_ae1?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: Clight\_ae1

Content-Type: application/json;ty=3

X-M2M-RI: mncse-13345

{

"m2m:cnt":

{

"rn": "light"

}

}

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-13345

Content-Location: /mn-cse/cnt-582759912

#### 9.7.5.2 Create a container of ADN-AE2

The creation of a container resource for ADN-AE2 is shown in the following procedure.

The following example shows a container create request and response using HTTP with XML serialization:

HTTP Request:

POST /home\_gateway/light\_ae2?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: Clight\_ae2

Content-Type: application/xml;ty=3

X-M2M-RI: mncse-62345

<m2m:cnt xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="light">

</m2m:cnt>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-62345

Content-Location: /mn-cse/cnt-582769893

The following example shows a container create request and response using HTTP with JSON serialization:

HTTP Request:

POST /home\_gateway/light\_ae2?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: Clight\_ae2

Content-Type: application/json;ty=3

X-M2M-RI: mncse-62345

{

"m2m:cnt":

{

"rn": "light"

}

}

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-62345

Content-Location: /mn-cse/cnt-582769893

### 9.7.6 ContentInstances creation

#### 9.7.6.1 Create a content instance of ADN-AE1

The creation of a content instance resource under the light container of ADN-AE1 with initial content OFF is shown in the following procedure.

The following example shows a contentInstance create request and response using HTTP with XML serialization:

HTTP Request:

POST /home\_gateway/light\_ae1/light?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: Clight\_ae1

Content-Type: application/xml;ty=4

X-M2M-RI: mncse-24345

<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols">

<cnf>text/plain:0</cnf>

<con>OFF</con>

</m2m:cin>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-24345

Content-Location: /mn-cse/cin-394798749

Content-Type: application/xml

The following example shows a contentInstance create request and response using HTTP with JSON serialization:

HTTP Request:

POST /home\_gateway/light\_ae1/light?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: Clight\_ae1

Content-Type: application/json;ty=4

X-M2M-RI: mncse-24345

{

"m2m:cin":

{

"cnf": "text/plains:0",

"con": "OFF"

}

}

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-24345

Content-Location: /mn-cse/cin-394798749

Content-Type: application/json

#### 9.7.6.2 Create a content instance of ADN-AE2

The creation of a content instance resource under the light container of ADN-AE2 with initial content OFF is shown in the following procedure.

The following example shows a contentInstance create request and response using HTTP with XML serialization.

HTTP Request:

POST /home\_gateway/light\_ae2/light?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: Clight\_ae1

Content-Type: application/xml;ty=4

X-M2M-RI: mncse-22345

<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols">

<cnf>text/plain:0</cnf>

<con>OFF</con>

</m2m:cin>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-22345

Content-Location: /mn-cse/cin-256599578

The following example shows a contentInstance create request and response using HTTP with JSON serialization:

HTTP Request:

POST /home\_gateway/light\_ae2/light?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: Clight\_ae1

Content-Type: application/json;ty=4

X-M2M-RI: mncse-22345

{

"m2m:cin":

{

"cnf": "text/plains:0",

"con": "OFF"

}

}

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-22345

Content-Location: /mn-cse/cin-256599578

### 9.7.7 Group creation

The creation of a group resource by the MN-AE is shown in the following procedure. The group resource is created with two initial member ids of the light container resources.

The following example shows a group create request and response using HTTP with XML serialization:

HTTP Request:

POST /home\_gateway/gateway\_ae?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: Cgateway\_ae

Content-Type: application/xml;ty=9

X-M2M-RI: mncse-76905

<?xml version="1.0" encoding="UTF-8"?>

<m2m:grp xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="containers\_grp">

<mt>3</mt>

<mid>/mn-cse/cnt-582759912 /mn-cse/cnt-582769893</mid>

<mnm>10</mnm>

</m2m:grp>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-76905

Content-Location: /mn-cse/grp-977978327

The following example shows a group create request and response using HTTP with JSON serialization:

HTTP Request:

POST /home\_gateway/gateway\_ae?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: Cgateway\_ae

Content-Type: application/json;ty=9

X-M2M-RI: mncse-76905

{

"m2m:grp":

{

"rn":"containers\_grp",

"mt": 3,

"mid":["/mn-cse/cnt-582759912", "/mn-cse/cnt-582769893"],

"mnm": 10

}

}

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-76905

Content-Location: /mn-cse/grp-977978327

### 9.7.8 Subscriptions creation

#### 9.7.8.1 Subscription to the content instance of ADN-AE1

When a subscription resource is created, the *notification content type* (*nct*) parameter is set to a value 1 to indicate that all attributes of the subscribed resource will be notified to the subscriber.

ADN-AE1 creates a subscription resource including the notification URI set to the resource identifier of ADN-AE1 so that the ADN-AE1 will get notified whenever a content instance child resource is created in the container. The corresponding subscription create request is shown in the following procedure.

The following example shows a subscription create request and response using HTTP with XML serialization:

HTTP Request:

POST /home\_gateway/light\_ae1/light?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: Clight\_ae1

Content-Type: application/xml;ty=23

X-M2M-RI: mncse-67891

<?xml version="1.0" encoding="UTF-8"?>

<m2m:sub xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="lightstate\_sub1">

<enc>

<net>3</net>

</enc>

<nu>Clight\_ae1</nu>

<nct>1</nct>

</m2m:sub>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-67891

Content-Location: /mn-cse/sub-856593979

The following example shows a subscription create request and response using HTTP with JSON serialization:

HTTP Request:

POST /home\_gateway/light\_ae1/light?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: Clight\_ae1

Content-Type: application/json;ty=23

X-M2M-RI: mncse-67891

{

"m2m:sub":

{

"rn": "lightstate\_sub1",

"enc":

{

"net":[3]

},

"nu":["Clight\_ae1"],

"nct":1

}

}

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-67891

Content-Location: /mn-cse/sub-856593979

#### 9.7.8.2 Subscription to the content instance of ADN-AE2

When a subscription resource is created, the *notification content type* (*nct*) parameter is set to a value 1 to indicate that all attributes of the subscribed resource will be notified to the subscriber.

ADN-AE1 creates a subscription resource including the notification URI set to the resource identifier of ADN-AE1 so that the ADN-AE1 will get notified whenever a content instance child resource is created in the container. The corresponding subscription create request is shown in the following procedure.

When a subscription resource is created, the *notification content type* (*nct*) parameter is set to value 1 to indicate that all attributes of the subscribed resource will be notified to the subscriber.

ADN-AE2 creates a subscription resource including the notification URI set to the resource identifier of ADN-AE2 so that the ADN-AE2 will get notified whenever a content instance child resource is created in the container. The corresponding subscription create request is shown in the following procedures.

The following example shows a subscription create request and response using HTTP with XML serialization:

HTTP Request:

POST /home\_gateway/light\_ae2/light?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: Clight\_ae2

Content-Type: application/xml;ty=23

X-M2M-RI: mncse-29387

<?xml version="1.0" encoding="UTF-8"?>

<m2m:sub xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="lightstate\_sub2">

<enc>

<net>3</net>

</enc>

<nu>Clight\_ae2</nu>

<nct>1</nct>

</m2m:sub>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-29387

Content-Location: /mn-cse/sub-856463728

The following example shows a subscription create request and response using HTTP with JSON serialization:

HTTP Request:

POST /home\_gateway/light\_ae2/light?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: Clight\_ae2

Content-Type: application/json;ty=23

X-M2M-RI: mncse-29387

{

"m2m:sub":

{

"rn": "lightstate\_sub2",

"enc":

{

"net":[3]

},

"nu":["Clight\_ae2"],

"nct":1

}

}

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-29387

Content-Location: /mn-cse/sub-856463728

### 9.7.9 Discovery

#### 9.7.9.1 Introduction

The discovery functionality in oneM2M is implemented using a RETRIEVE operation along with one or multiple filter criteria parameters.

In order to enable the retrieve operation for resource discovery, parameter *filterUsage* (*fu)* is included in the RETRIEVE request as a query string.

In addition, parameter *resource type* (*rty)* is used as a *filterCriteria* condition for the discovery of single light and group light members. The parameter *discovery result type* (*drt*)is set to 2 to indicate that the format of elements of URIList is unstructured. The detailed discovery procedures are presented in clauses 9.7.9.2 and 9.7.9.3.

#### 9.7.9.2 Discovery of single light registered with MN-CSE

The discovery of containers for each light registered with the MN-CSE by the smartphone AE is shown in the following procedure.

If the discovery response is preferred to be returned with a XML representation, the HTTP request message is sent as following example:

HTTP Request:

GET /~/mn-cse/home\_gateway?fu=1&rty=3&drt=2 HTTP/1.1

Host: in.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

X-M2M-RI: mncse-99882

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-99882

X-M2M-CNST: 2

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<m2m:uril xmlns:m2m="http://www.onem2m.org/xml/protocols">

/mn-cse/cnt-582759912

/mn-cse/cnt-582769893

</m2m:uril>

If the discovery response is preferred to be returned with a JSON representation, the HTTP request message is sent as following example:

HTTP Request:

GET /~/mn-cse/home\_gateway?fu=1&rty=3&drt=2 HTTP/1.1

Host: in.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

X-M2M-RI: mncse-99882

Accept: application/json

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-99882

X-M2M-CNST: 2

Content-Type: application/json

{

"m2m:uril":

[

"/mn-cse/cnt-582759912",

"/mn-cse/cnt-582769893"

]

}

The smartphone application retrieves a list of URIs representing containers registered with MN-CSE from the response message, e.g., /mn-cse/cnt-582759912 which is the URI of container created in ADN-AE1. The retrieved URIs of the discovered containers are then used for the group member update operation.

#### 9.7.9.3 Discovery of groups located in MN-CSE

The discovery of groups located in MN-CSE by the smartphone AE is shown in the following procedures.

If the discovery response is preferred to be returned with a XML representation, the HTTP request message is sent as following example:

HTTP Request:

GET /~/mn-cse/home\_gateway?fu=1&rty=9&drt=2 HTTP/1.1

Host: in.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

X-M2M-RI: mncse-15001

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-15001

X-M2M-CNST: 2

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<m2m:uril xmlns:m2m="http://www.onem2m.org/xml/protocols">

/mn-cse/grp-977978327

</m2m:uril>

If the discovery response is preferred to be returned with a JSON representation, the HTTP request message is sent as following example:

HTTP Request:

GET /~/mn-cse/home\_gateway?fu=1&rty=9&drt=2 HTTP/1.1

Host: in.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

X-M2M-RI: mncse-15001

Accept: application/json

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-15001

X-M2M-CNST: 2Content-Type: application/json

{

"m2m:uril":

[

"/mn-cse/grp-977978327"

]

}

The smartphone application retrieves a list of URIs representing group resources located in MN-CSE from the response message, e.g., /mn-cse/grp-977978327 which is the URI of the group resource. The retrieved URIs of the discovered group resource are then used for the group member update operation.

### 9.7.10 Latest content instances retrieval

#### 9.7.10.1 Introduction

The smartphone application can retrieve the latest light states via sending a RETRIEVE request targeting a container's *la* (short for *latest*) virtual resource*.*

The smartphone application can also retrieve a group of latest light states via sending a RETRIEVE request targeting the group *fanOutPoint* virtual resource.

#### 9.7.10.2 Retrieve the latest content instance of ADN-AE1

The latest content instance of the *light* container resource for ADN-AE1 can be retrieved by the following procedure.

If the response is preferred to be returned with a XML representation, the following is a HTTP request message example:

HTTP Request:

GET /~/mn-cse/home\_gateway/light\_ae1/light/la HTTP/1.1

Host: in.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

X-M2M-RI: mncse-11223

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-11223

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<m2m:cin

xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="cin-394798749">

<ty>4</ty>

<ri>cin-394798749</ri>

<pi>cnt-181049109</pi>

<ct>20150925T045938</ct>

<lt>20150925T045938</lt>

<et>20151107T154802</et>

<st>0</st>

<cnf>text/plain:0</cnf>

<cs>3</cs>

<con>OFF</con>

</m2m:cin>

If the response is preferred to be returned with a JSON representation, the following is a HTTP request message example:

GET /~/mn-cse/home\_gateway/light\_ae1/light/lat HTTP/1.1

Host: in.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

X-M2M-RI: mncse-11223

Accept: application/json

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-11223

Content-Type: application/json

{

"m2m:cin":

{

"ty":4,

"ri":"cin-394798749",

"pi":"cnt-181049109",

"rn":"cin-394798749",

"ct":"20150925T045938",

"lt":"20150925T045938",

"et":"20151107T154802",

"st":0,

"cnf":"text/plain:0",

"cs":3,

"con":"OFF"

}

}

#### 9.7.10.3 Retrieve the latest content instance of ADN-AE2

The latest content instance of the *light* container resource for ADN-AE2 can be retrieved by the following procedures.

If the response is preferred to be represented in XML, the following is a HTTP request message example:

HTTP Request:

GET /~/mn-cse/home\_gateway/light\_ae2/light/la HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

X-M2M-RI: mncse-22336

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-22336

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<m2m:cin

xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="cin-256599578">

<ty>4</ty>

<ri>cin-256599578</ri>

<pi>cnt-790965889</pi>

<ct>20150925T050515</ct>

<lt>20150925T050515</lt>

<et>20151107T154802</et>

<st>0</st>

<cnf>text/plain:0</cnf>

<cs>3</cs>

<con>OFF</con>

</m2m:cin>

If the response is preferred be returned in representation of JSON, the following is a HTTP request message example:

GET /~/mn-HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

X-M2M-RI: mncse-22336

Accept: application/json

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-22336

Content-Type: application/json

{

"m2m:cin":

{

"ty":4,

"ri": "cin-256599578",

"pi": "cnt-790965889",

"rn": "cin-256599578",

"ct": "20150925T050515",

"lt": "20150925T050515",

"et": "20151107T154802",

"st": 0,

"cnf": "text/plain:0",

"vcs": 3,

"con": "OFF"

}

}

#### 9.7.10.4 Retrieve a group of latest content instances for all light states

A group of latest content instances can be retrieved via sending a RETRIEVE request targeting the group *fanOutPoint* virtual resource and appending *latest* as shown in the following procedures.

If the response is preferred to be returned with a XML representation, the following is a HTTP request message example:

HTTP Request:

GET /~/mn-cse/home\_gateway/gateway\_ae/containers\_grp/fopt/la HTTP/1.1

Host: in.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

X-M2M-RI: mncse-55667

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-55667

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<m2m:agr

xmlns:m2m="http://www.onem2m.org/xml/protocols">

<m2m:rsp>

<rsc>2000</rsc>

<rqi>mncse-55667</rqi>

<pc>

<m2m:cin rn="cin-394798749">

<ty>4</ty>

<ri>cin-394798749</ri>

<pi>cnt-181049109</pi>

<ct>20150925T045938</ct>

<lt>20150925T045938</lt>

<et>20151107T154802</et>

<st>0</st>

<cnf>text/plain:0</cnf>

<cs>3</cs>

<con>OFF</con>

</m2m:cin>

</pc>

<to>/in-cse/Csmartphone\_ae</to>

<fr>/mn-cse/cnt-582759912/la</fr>

</m2m:rsp>

<m2m:rsp>

<rsc>2000</rsc>

<rqi>mncse-55667</rqi>

<pc>

<m2m:cin rn="cin-256599578">

<ty>4</ty>

<ri>cin-256599578</ri>

<pi>cnt-790965889</pi>

<ct>20150925T050515</ct>

<lt>20150925T050515</lt>

<et>20151107T154802</et>

<st>0</st>

<cnf>text/plain:0</cnf>

<cs>3</cs>

<con>OFF</con>

</m2m:cin>

</pc>

<to>/in-cse/Csmartphone\_ae</to>

<fr>/mn-cse/cnt-582769893/la</fr>

</m2m:rsp>

</m2m:agr>

If the response is preferred to be returned with a JSON representation, the following is a HTTP request message example:

HTTP Request:

GET /~/mn-cse/home\_gateway/containers\_grp/fopt/la HTTP/1.1

Host: in.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

X-M2M-RI: mncse-55667

Accept: application/json

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-55667

Content-Type: application/json

{

"m2m:agr":

{

"m2m:rsp":[

{

"rsc": 2000,

"rqi": "mncse-55667",

"pc":

{

"m2m:cin":

{

"ty":4,

"ri": "cin-394798749",

"pi": "cnt-181049109",

"rn": "cin-394798749",

"ct": "20150925T045938",

"lt": "20150925T045938",

"et": "20151107T154802",

"st": 0,

"cnf":"text/plain:0",

"cs": 3,

"con": "OFF"

}

},

"to":"/in-cse/Csmartphone\_ae",

"fr":"/mn-cse/cnt-582759912/la"

},

{

"rsc":2000,

"rqi": "mncse-55667",

"pc":

{

"m2m:cin":

{

"ty":4,

"ri":"cin-256599578",

"pi":"cnt-790965889",

"rn":"cin-256599578",

"ct":"20150925T050515",

"lt":"20150925T050515",

"et":"20151107T154802",

"st":0,

"cnf":"text/plain:0",

"cs":3,

"con":"OFF"

}

},

"to":"/in-cse/Csmartphone\_ae",

"fr":"/mn-cse/cnt-582769893/la"

}

]}

}

### 9.7.11 Light state modification

#### 9.7.11.1 Introduction

Once the smartphone application is registered with the IN-CSE, it can be granted access to resources including containers located in the MN-CSE so that smartphone application users can send light control commands for modifying the light states.

When the user makes a change to the light state via the smartphone user interface, the smartphone application performs a new content instance creation procedure carrying the new state.

The modification of a single light state is implemented by creating a new content instance resource for the specific container with ACP acp1 (*gateway\_acp*) while the modification of all light states is implemented by creating a new content instance resource for each member of group (*containers\_grp*) with ACP acp1 (*gateway\_acp*). The implementation of the latter case is to target the <fopt> virtual resource of *containers\_grp* resource with a content instance create request so that the content of all members of the group is updated together.

#### 9.7.11.2 Create a content instance under container of ADN-AE1

If the contentInstance create request body is represented in XML, the following is a HTTP request message example:

HTTP Request:

POST /~/mn-cse/home\_gateway/light\_ae1/light?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

Content-Type: application/xml;ty=4

X-M2M-RI: mncse-11123

<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols">

<cnf>text/plain:0</cnf>

<con>ON</con>

</m2m:cin>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-11123

Content-Location: /mn-cse/cin-789356234

If the contentInstance create request body is represented in JSON, the following is a HTTP request message example:

HTTP Request:

POST /~/mn-cse/home\_gateway/light\_ae1/light?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

Content-Type: application/json;ty=4

X-M2M-RI: mncse-11123

{

"m2m:cin":

{

"cnf": "text/plains:0",

"con": "ON"

}

}

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-11123

Content-Location: /mn-cse/cin-789356234

#### 9.7.11.3 Create a content instance under container of ADN-AE2

If the contentInstance create request body is represented in XML, the following is an HTTP request message the example:

HTTP Request:

POST /~/mn-cse/home\_gateway/light\_ae2/light?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

Content-Type: application/xml;ty=4

X-M2M-RI: mncse-12222

<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols">

<cnf>text/plain:0</cnf>

<con>ON</con>

</m2m:cin>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse/12222

Content-Location: /mn-cse-cin-237896783

If the contentInstance create request body is represented in JSON, the following is a HTTP request message example:

HTTP Request:

POST /~/mn-cse/home\_gateway/light\_ae2/light?rcn=0 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

Content-Type: application/json;ty=4

X-M2M-RI: mncse-12222

{

"m2m:cin":

{

"cnf": "text/plains:0",

"con": "ON"

}

}

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: mncse-12222

Content-Location: /mn-cse/cin-237896783

#### 9.7.11.4 Update the state of all lights using group fanout

If the fanOutPoint request body is represented in XML, the following is a HTTP request message example:

HTTP Request:

POST /~/mn-cse/home\_gateway/gateway\_ae/containers\_grp/fopt HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

Content-Type: application/xml

X-M2M-RI: mncse-33344

<?xml version="1.0" encoding="UTF-8"?>

<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols">

<cnf>text/plain:0</cnf>

<con>ON</con>

</m2m:cin>

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-33344

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<m2m:agr

xmlns:m2m="http://www.onem2m.org/xml/protocols">

<m2m:rsp>

<rsc>2001</rsc>

<rqi>mncse-33344</rqi>

<pc>

<m2m:cin rn="cin-479874939">

<ty>4</ty>

<ri>cin-479874939</ri>

<pi>cnt-181049109</pi>

<ct>20151025T045938</ct>

<lt>20151025T045938</lt>

<et>20151207T154802</et>

<st>0</st>

<cs>2</cs>

</m2m:cin>

</pc>

<to>/in-cse/Csmartphone\_ae</to>

<fr>/mn-cse/cnt-582759912</fr>

</m2m:rsp>

<m2m:rsp>

<rsc>2001</rsc>

<rqi>mncse-33344</rqi>

<pc>

<m2m:cin rn="cin-659957825">

<ty>4</ty>

<ri>cin-659957825</ri>

<pi>cnt-790965889</pi>

<ct>20151025T045938</ct>

<lt>20151025T045938</lt>

<et>20151207T154802</et>

<st>0</st>

<cs>2</cs>

</m2m:cin>

</pc>

<to>/in-cse/Csmartphone\_ae</to>

<fr>/mn-cse/cnt-582769893</fr>

</m2m:rsp>

</m2m:agr>

If the fanOutPoint request body is represented in JSON, the following is a HTTP request message example:

HTTP Request:

POST /~/mn-cse/home\_HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

Content-Type: application/json

X-M2M-RI: mncse-33344

{

"m2m:cin":

{

"cnf": "text/plains:0",

"con": "ON"

}

}

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-33344

Content-Type: application/json

{

"m2m:agr":

{

"m2m:rsp":

[{

"rsc": 2001,

"rqi":"mncse-33344",

"pc":

{

"m2m:cin":

{

"ty": 4,

"ri": "cin-479874939",

"pi": "cnt-181049109",

"rn": "cin-479874939",

"ct": "20151025T045938",

"lt": "20151025T045938",

"et": "20151207T154802",

"st": 0,

"cs": 2

}

},

"to": "/in-cse/Csmartphone\_ae",

"fr": "/mn-cse/cnt-582759912"

},

{

"rsc": 2001,

"rqi": "mncse-33344",

"pc":

{

"m2m:cin":

{

"ty": 4,

"ri": "cin-659957825",

"pi": "cnt-790965889",

"rn": "cin-659957825",

"ct": "20151025T050515",

"lt": "20151025T050515",

"et": "20151207T154802",

"st": 0,

"cs": 2

}

},

"to": "/in-cse/Csmartphone\_ae ",

"fr": "/mn-cse/cnt-582769893"

}

]}

}

### 9.7.12 Notifications

#### 9.7.12.1 Introduction

Each time a content instance is created under a container of an ADN-AE, then a notification containing the whole created content instance is posted to the targeted subscriber i.e., ADN-AE1 or ADN-AE2, that can actuate the light with the new state received in the notification.

#### 9.7.12.2 Post a notification to ADN-AE1

If the notification request body is represented in XML, the following is a HTTP request message example:

HTTP Request:

POST / HTTP/1.1

Host: 192.168.0.10:9090

X-M2M-Origin: /mn-cse

X-M2M-RI: notif-12345

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<m2m:sgn

xmlns:m2m="http://www.onem2m.org/xml/protocols">

<nev>

<rep>

<m2m:cin>

<cnf>text/plain:0</cnf>

<con>ON</con>

</m2m:cin>

</rep>

<net>3</net>

</nev>

<sur>

/mn-cse/sub-856593979

</sur>

</m2m:sgn>

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: notif-12345

If the notification request body is represented in JSON, the following is a HTTP request message:

HTTP Request:

POST / HTTP/1.1

Host: 192.168.0.10:9090

X-M2M-Origin: /mn-cse

X-M2M-RI: notif-12345

Content-Type: application/json

{

"m2m:sgn":

{

"nev":{

"rep":

{

"cin":

{

"cnf": "text/plain:0" ,

"con": "ON"

}

},

"net":[3]

},

"sur":"/mn-cse/sub-856593979"

}

}

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: notif-12345

#### 9.7.12.3 Post a notification to ADN-AE2

If the notification request body is represented in XML, the following is a HTTP request message example:

HTTP Request:

POST / HTTP/1.1

Host: 192.168.0.20:9090

X-M2M-Origin: /mn-cse

X-M2M-RI: notif-12346

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<m2m:sgn

xmlns:m2m="http://www.onem2m.org/xml/protocols">

<nev>

<rep>

<cin>

<cnf>text/plain:0</cnf>

<con>ON</con>

</cin>

</rep>

<net>3</net>

</nev>

<sur>

/mn-cse/sub-856463728

</sur>

</m2m:sgn>

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: notif-12346

If the notification request body is represented in JSON, the following is a HTTP request message example:

HTTP Request:

POST / HTTP/1.1

Host: 192.168.0.20:9090

X-M2M-Origin: /mn-cse

X-M2M-RI: notif-12346

Content-Type: application/json

{

"m2m:sgn":

{

"nev":{

"rep":

{

"cin":

{

"cnf": "text/plain:0" ,

"con": "ON"

}

},

"net":[3]

},

"sur":"/mn-cse/sub-856463728"

}

}

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: notif-12346

Appendix A  
  
Reading resources

## A.1 Introduction

The information of resources created during each call flow of current use case can be retrieved via smartphone application IN-AE initiates a RETRIEVE request to the target resources as following.

## A.2 CSE resources

### A.2.1 IN-CSE

HTTP Request:

GET /~/in-cse/server HTTP/1.1

Host: in.provider.com:8080

X-M2M-Origin: /mn-cse/Cgateway\_ae

X-M2M-RI: incse-12345

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: incse-12345

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<m2m:cb xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="server">

<ty>5</ty>

<ri>/in-cse</ri>

<ct>20150925T045938</ct>

<lt>20150925T045938</lt>

<acpi>/in-cse/acp-666957710</acpi>

<cst>1</cst>

<csi>in-cse</csi>

<srt>1 2 3 4 5 9 14 15 16 17 23</srt>

<poa>http://in.provider.com:8080/</poa>

</m2m:cb>

### A.2.2 MN-CSE

HTTP Request:

GET /~/in-cse/server/home\_gateway HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

X-M2M-RI: incse-12346

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: incse-12346

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<m2m:csr xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="home\_gateway">

<ty>16</ty>

<ri>csr-299409504</ri>

<pi>/in-cse</pi>

<ct>20150925T045938</ct>

<lt>20150925T045938</lt>

<et>20171005T105550</et>

<acpi>/in-cse/acp-666957710</acpi>

<poa>http://mn.provider.com:8080/</poa>

<cb>mn.provider.com/mn-cse</cb>

<csi>/mn-cse</csi>

<rr>true</rr>

</m2m:csr>

## A.3 Gateway device application MN-AE

HTTP Request:

GET /~/mn-cse/home\_gateway/gateway\_ae HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

X-M2M-RI: mncse-12347

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-12347

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="gateway\_ae">

<ty>2</ty>

<ri>ae-CAE340303271</ri>

<pi>/mn-cse</pi>

<ct>20150925T052438</ct>

<lt>20150925T052438</lt>

<et>20171005T105550</et>

<acpi>/mn-cse/acp-805496226</acpi>

<api>A01.com.company.gatewayApp</api>

<aei>CAE340303271</aei>

<rr>false</rr>

</m2m:ae>

## A.4 Light device applications

### A.4.1 ADN-AE1

HTTP Request:

GET /~/mn-cse/home\_gateway/light\_ae1 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /mn-cse/Cgateway\_ae

X-M2M-RI: mncse-12347

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-12347

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="light\_ae1">

<ty>2</ty>

<ri>ae-CAE340304071</ri>

<pi>/mn-cse</pi>

<ct>20150925T052455</ct>

<lt>20150925T052455</lt>

<et>20171005T105550</et>

<acpi>/mn-cse/acp-805496226</acpi>

<api>A01.com.company.lightApp1</api>

<aei>CAE340304071</aei>

<rr>true</rr>

</m2m:ae>

### A.4.2 ADN-AE2

HTTP Request:

GET /~/mn-cse/home\_gateway/light\_ae2 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /mn-cse/Cgateway\_ae

X-M2M-RI: mncse-12348

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-12348

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="light\_ae2">

<ty>2</ty>

<ri>ae-CAE340304042</ri>

<pi>/mn-cse</pi>

<ct>20150925T052542</ct>

<lt>20150925T052542</lt>

<et>20171005T105550</et>

<acpi>/mn-cse/acp-805496226</acpi>

<api>A01.com.company.lightApp2</api>

<aei>CAE340304042</aei>

<rr>true</rr>

</m2m:ae>

## A.5 Smartphone application IN-AE

HTTP Request:

GET /~/in-cse/server/smartphone\_ae HTTP/1.1

Host: in.provider.com:8080

X-M2M-Origin: /mn-cse/Cgateway\_ae

X-M2M-RI: incse-12349

Accept: application/xml

Resonse status:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: incse-12349

Content-Type: application/xml

Resonse message:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="smartphone\_ae">

<ty>2</ty>

<ri>ae-CAE340304178</ri>

<pi>/in-cse</pi>

<ct>20150925T052622</ct>

<lt>20150925T052622</lt>

<et>20171005T105550</et>

<acpi>/in-cse/acp-666957710</acpi>

<api>A01.com.company.lightControlApp</api>

<aei>CAE340304178</aei>

<rr>false</rr>

</m2m:ae>

## A.6 Access control policy

HTTP Request:

GET /~/mn-cse/home\_gateway/gateway\_acp HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /mn-cse/Cgateway\_ae

X-M2M-RI: mncse-12350

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-12350

Content-Type: application/xml

<m2m:acp xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="gateway\_acp">

<ty>1</ty>

<ri>acp-805496226</ri>

<pi>/mn-cse</pi>

<ct>20150925T050238</ct>

<lt>20150925T050238</lt>

<et>20171005T105550</et>

<pv>

<acr>

<acor>/in-cse/home\_gateway /mn-cse/Cgateway\_ae /mn-cse/Clight\_ae1 /mn-cse/Clight\_ae2 /in-cse/Csmartphone\_ae</acor>

<acop>63</acop>

</acr>

</pv>

<pvs>

<acr>

<acor>/in-cse/home\_gateway /mn-cse/Cgateway\_ae /mn-cse/Clight\_ae1 /mn-cse/Clight\_ae2 /in-cse/Csmartphone\_ae</acor>

<acop>51</acop>

</acr>

</pvs>

</m2m:acp>

## A.7 Containers

### A.7.1 Container under ADN-AE1

HTTP Request:

GET /~/mn-cse/home\_gateway/light\_ae1/light HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /mn-cse/Cgateway\_ae

X-M2M-RI: mncse-12351

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-12351

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<m2m:cnt xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="light">

<ty>3</ty>

<ri>cnt-582759912</ri>

<pi>/mn-cse/ae-CAE340304071</pi>

<ct>20150925T052955</ct>

<lt>20150925T052955</lt>

<et>20171005T105550</et>

<acpi>/mn-cse/acp-805496226</acpi>

<st>0</st>

<cni>1</cni>

<cbs>3</cbs>

</m2m:cnt>

### A.7.2 Container under ADN-AE2

HTTP Request:

GET /~/mn-cse/home\_gateway/light\_ae2/light HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /mn-cse/Cgateway\_ae

X-M2M-RI: mncse-12352

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-12352

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<m2m:cnt xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="light">

<ty>3</ty>

<ri>cnt-582769893</ri>

<pi>/mn-cse/ae-CAE340304042</pi>

<ct>20150925T053135</ct>

<lt>20150925T053135</lt>

<et>20171005T105550</et>

<acpi>/mn-cse/acp-805496226</acpi>

<st>0</st>

<cni>1</cni>

<cbs>3</cbs>

</m2m:cnt>

## A.8 ContentInstances

### A.8.1 Latest contentInstance in ADN-AE1

HTTP Request:

GET /~/mn-cse/home\_gateway/light\_ae1/light/la HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /mn-cse/Cgateway\_ae

X-M2M-RI: mncse-12353

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-12353

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<m2m:cin

xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="cin-394798749">

<ty>4</ty>

<ri>cin-394798749</ri>

<pi>/mn-cse/cnt-582759912</pi>

<ct>20150925T053225</ct>

<lt>20150925T053225</lt>

<et>20171005T105550</et>

<st>0</st>

<cnf>text/plain:0</cnf>

<cs>2</cs>

<con>ON</con>

</m2m:cin>

### A.8.2 Latest contentInstance in ADN-AE2

HTTP Request:

GET /~/mn-cse/home\_gateway/light\_ae2/light/la HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /mn-cse/Cgateway\_ae

X-M2M-RI: mncse-12354

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-12354

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<m2m:cin

xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="cin-256599578">

<ty>4</ty>

<ri>cin-256599578</ri>

<pi>/mn-cse/cnt-582769893</pi>

<ct>20150925T053425</ct>

<lt>20150925T053425</lt>

<et>20171005T105550</et>

<st>0</st>

<cnf>text/plain:0</cnf>

<cs>2</cs>

<con>ON</con>

</m2m:cin>

## A.9 Subscriptions

### A.9.1 Subscription to container in ADN-AE1

HTTP Request:

GET /~/mn-cse/home\_gateway/light\_ae1/light/lightstate\_sub1 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /mn-cse/Cgateway\_ae

X-M2M-RI: mncse-12355

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-12355

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<m2m:sub xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="lightstate\_sub1">

<ty>23</ty>

<ri>sub-856593979</ri>

<pi>/mn-cse/cnt-582759912</pi>

<ct>20150926T052955</ct>

<lt>20150926T052955</lt>

<et>20171005T105550</et>

<acpi>/mn-cse/acp-805496226</acpi>

<enc>

<net>3</net>

</enc>

<nu>/mn-cse/ae-CAE340304071</nu>

<nct>1</nct>

</m2m:sub>

### A.9.2 Subscription to container in ADN-AE2

HTTP Request:

GET /~/mn-cse/home\_gateway/light\_ae2/light/lightstate\_sub2 HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /mn-cse/Cgateway\_ae

X-M2M-RI: mncse-12356

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-12356

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<m2m:sub xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="lightstate\_sub2">

<ty>23</ty>

<ri>sub-856463728</ri>

<pi>/mn-cse/cnt-582759912</pi>

<ct>20150926T053055</ct>

<lt>20150926T053055</lt>

<et>20171005T105550</et>

<acpi>/mn-cse/acp-805496226</acpi>

<enc>

<net>3</net>

</enc>

<nu>/mn-cse/ae-CAE340304042</nu>

<nct>1</nct>

</m2m:sub>

## A.10 Groups

### A.10.1 Group1

HTTP Request:

GET /~/mn-cse/home\_gateway/gateway\_ae/containers\_grp HTTP/1.1

Host: mn.provider.com:8080

X-M2M-Origin: /in-cse/Csmartphone\_ae

X-M2M-RI: mncse-12357

Accept: application/xml

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: mncse-12357

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<m2m:grp xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="containers\_grp">

<ty>9</ty>

<ri>grp-977978327</ri>

<pi>/mn-cse/ae-CAE340303271</pi>

<ct>20151004T045954</ct>

<lt>20151004T045954</lt>

<et>20171005T105550</et>

<acpi>/mn-cse/acp-805496226 /mn-cse/acp-805496226</acpi>

<mt>3</mt>

<cnm>2</cnm>

<mnm>10</mnm>

<mid>/mn-cse/cnt-582759912 /mn-cse/cnt-582769893</mid>

<mtv>true</mtv>

<csy>1</csy>

<fopt>/mn-cse/grp-977978327/fopt</fopt>

</m2m:grp>

\_\_\_\_\_\_\_\_\_\_\_\_\_