



ONEM2M TECHNICAL REPORT

Document Number	TR-0035-V-2.0.0
Document Name:	Developer guide of device management
Date:	2018-03-12
Abstract:	The document provides example of device management in oneM2M.

Template Version: 08 September 2015 (Dot not modify)

This Specification is provided for future development work within oneM2M only. The Partners accept no liability for any use of this Specification.

The present document has not been subject to any approval process by the oneM2M Partners Type 1. Published oneM2M specifications and reports for implementation should be obtained via the oneM2M Partners' Publications Offices.

About oneM2M

The purpose and goal of oneM2M is to develop technical specifications which address the need for a common M2M Service Layer that can be readily embedded within various hardware and software, and relied upon to connect the myriad of devices in the field with M2M application servers worldwide.

More information about oneM2M may be found at: <http://www.oneM2M.org>

Copyright Notification

© 2018, oneM2M Partners Type 1 (ARIB, ATIS, CCSA, ETSI, TIA, TSDSI, TTA, TTC).

All rights reserved.

The copyright and the foregoing restriction extend to reproduction in all media.

Notice of Disclaimer & Limitation of Liability

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. NO oneM2M PARTNER TYPE 1 SHALL BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY THAT PARTNER FOR THIS DOCUMENT, WITH RESPECT TO ANY CLAIM, AND IN NO EVENT SHALL oneM2M BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. oneM2M EXPRESSLY ADVISES ANY AND ALL USE OF OR RELIANCE UPON THIS INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

Contents

1	Scope	4
2	References	4
2.1	Normative references	4
2.2	Informative references	4
3	Abbreviations	4
4	Conventions.....	5
5	Device management over the service layer	5
5.1	Introduction.....	5
5.2	Use case	5
5.3	Architecture	6
5.3.1	Management of ASN, MN and IN	6
5.3.2	Management of ADN.....	7
5.3.3	Management of NoDN.....	7
5.4	Procedures.....	8
5.4.1	Management of ASN, MN and IN	8
5.4.2	Management of ADN.....	9
5.4.3	Management of NoDN.....	10
5.5	Implementation	10
5.5.1	Introduction.....	10
5.5.2	Management of ASN, MN and IN	11
5.5.3	Management of ADN.....	11
5.5.4	Management of NoDN.....	12
6	Device management using external management technologies	12
6.1	Introduction.....	12
6.2	Use case	12
6.3	Architecture	13
6.4	Procedures.....	14
6.5	Implementation	14
	History	16

1 Scope

The present document gives example of message flow and procedures on how device management will be performed using oneM2M defined primitives and procedures.

2 References

2.1 Normative references

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

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

Not applicable.

2.2 Informative references

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

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

[i.1] oneM2M Drafting Rules.

NOTE: Available at <http://www.onem2m.org/images/files/oneM2M-Drafting-Rules.pdf>.

[i.2] oneM2M TS-0001: "Functional Architecture".

[i.3] BBF TR-069: "Abstract Test Plan".

3 Abbreviations

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

ADN	Application Dedicated Node
ADN-AE	AE which resides in the Application Dedicated Node
AE	Application Entity
ASN	Application Service Node
BBF	Broadband Forum
CoAP	Constrained Application Protocol
CSE	Common Services Entity
CSE-ID	Common Service Entity Identifier
DM	Device Management
HTTP	HyperText Transfer Protocol
IN	Infrastructure Node
IN-CSE	CSE which resides in the Infrastructure Node
JSON	JavaScript Object Notation
Mca	Reference Point for M2M Communication with AE
MN	Middle Node
MN-CSE	CSE which resides in the Middle Node
NoDN	Non-oneM2M Node

OMA	Open Mobile Alliance
URI	Uniform Resource Identifier
URL	Uniform Resource Link

4 Conventions

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

5 Device management over the service layer

5.1 Introduction

oneM2M uses <mgmtObj> resource for device management. For device management over the service layer, the <mgmtObj> that is used for the management of the managed entity is located at different places in case of the characteristic of the managed entity. In either of the case, there exists one driver or client on the managed entity that observes the update of the <mgmtObj> resource to perform management operation.

For ASN, MN and IN which has CSE, the <mgmtObj> is hosted on the corresponding ASN-CSE, MN-CSE and IN-CSE as child resource of the <node> resource that represents the ASN, MN and IN. The managed entity observes the <mgmtObj> directly to perform the management operation.

For ADN, the <mgmtObj> resource is hosted on the registrar CSE of the ADN-AE as child resource of the <node> resource that represents the ADN. The ADN-AE in this case should subscribe the <mgmtObj> resources under the corresponding <node> resource to receive notifications on any update of the <mgmtObj> resource. The ADN which is the managed entity in this case further perform the management operation based on the notifications received.

For NoDN, the <mgmtObj> resource is hosted on the CSE which is NoDN is connected to. As NoDN is not an oneM2M defined entity, the connection between NoDN and the CSE is out of scope of oneM2M and is implementation specific. In this case, the <mgmtObj> resource on the CSE under the corresponding <node> resource should be monitored to determine if any management operation need to be performed. This part is implementation specific and is out of scope of oneM2M as well.

5.2 Use case

An application want to retrieve the current available memory of the device. In this case, the specialization of <mgmtObj> memory is used clause D.4 of [i.2]. In the memory specialization, *memAvailable* is the attribute that contains the required information. The application can get to know the available memory by retrieving the resource.

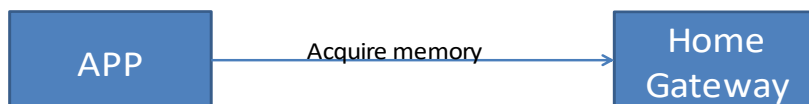


Figure 5.2-1: Use case for management

5.3 Architecture

5.3.1 Management of ASN, MN and IN

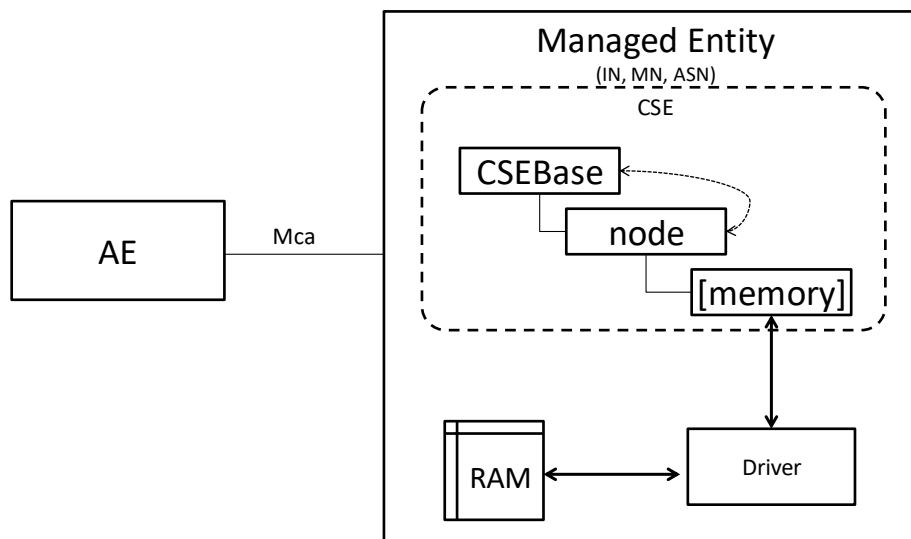


Figure 5.3.1-1: Architecture for management of IN,MN,ASN

In the architecture for management of IN, MN and ASN. As those Managed Entities have their own CSEBase resource, they have the capability to host oneM2M resources. When using device management over the service layer, the *[memory]* resource is directly hosting under the <node> resource of the CSEBase. In this case, the <node> resource represents the IN, MN, ASN themselves.

The Driver is the software that interacts with the memory of the device that acquires the total memory and available memory from the system.

In this case, the *[memory]* specialization is modified by Driver using internal interface. The modification may be done periodically or some other policies which is out of scope of oneM2M standard.

Whenever the AE issues retrieve to the *[memory]* specialization, the value from the resource is returned.

5.3.2 Management of ADN

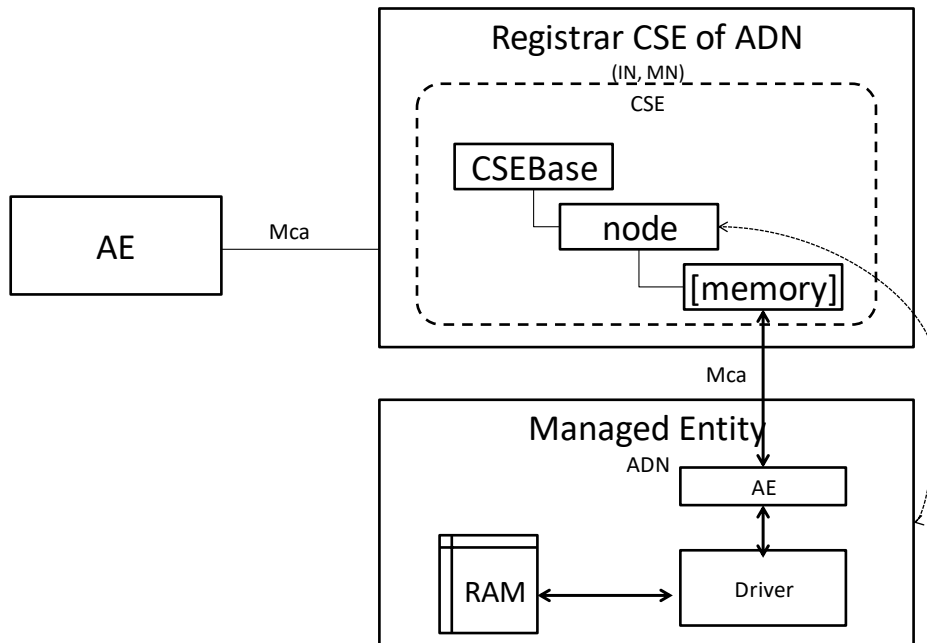


Figure 5.3.2-1: Architecture for management of ADN

In the architecture of management of ADN, the *[memory]* specialization is hosted on the Registrar CSE of the ADN. According to the supported configuration of oneM2M architecture, the Registrar CSE could be IN-CSE or MN-CSE.

In this case, the <node> resource hosted under the CSEBase of the Registrar CSE represents the ADN which is the managed entity. The driver gets the memory information using internal interfaces and make the ADN-AE updates the memory value to the *[memory]* specialization over the Mca reference point. The update over Mca is triggered periodically or based on the policy of the ADN which is implementation specific.

5.3.3 Management of NoDN

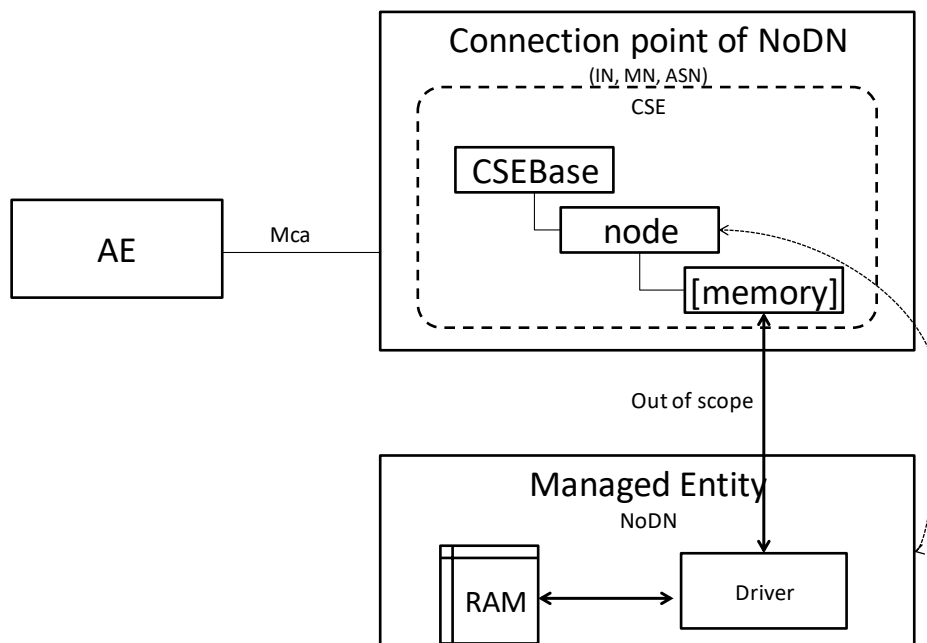


Figure 5.3.3-1: Architecture for management of NoDN

In the architecture of management of NoDN, the *[memory]* specialization is hosted on the connection point of NoDN. The connection point is the entity that the NoDN is connected to. The entity has the adaptor that shares the same network protocol with NoDN.

In this case, the <node> resource hosted under the CSEBase represents the NoDN. And the connection between Driver and the CSE is out of scope of oneM2M. It may be Bluetooth, ZigBee or even proprietary.

The update of the memory information happens over the interface between driver and CSE which is out of scope of oneM2M.

5.4 Procedures

5.4.1 Management of ASN, MN and IN

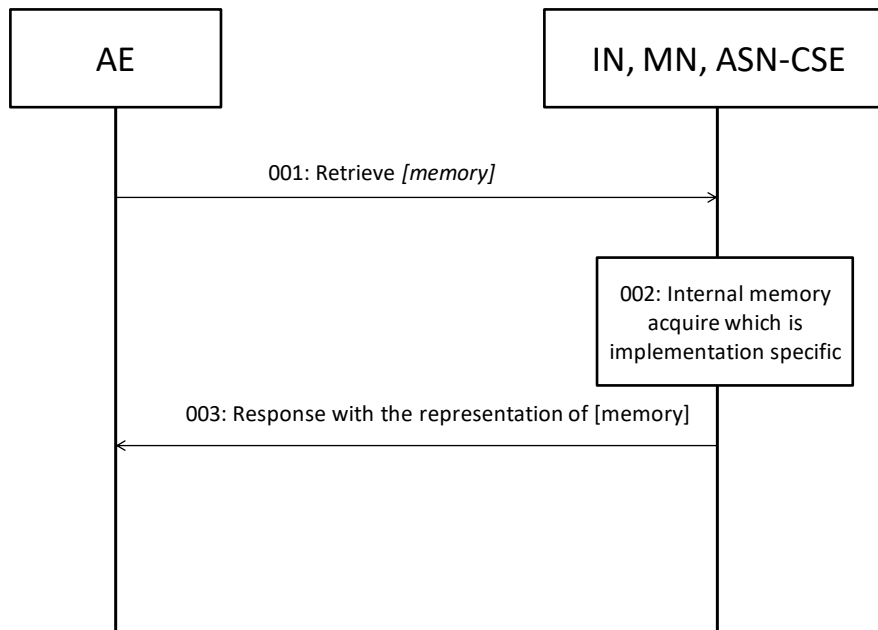


Figure 5.4.1-1: Procedures for management of IN, MN, ASN

001: The AE send an oneM2M Retrieve primitive to the ResourceID of *[memory]* resource.

002: The CSE receives the Retrieve primitive and triggers the Driver to acquire the current available memory value, update the *[memory]* resource.

003: The CSE responds the AE with the representation of the *[memory]* resource.

NOTE: The AE could also subscribe the *[memory]* resource to get the notification on the update of the *[memory]* resource.

The *[memory]* resource in this case is locally created before hand by the Hosting CSE using internal interfaces.

5.4.2 Management of ADN

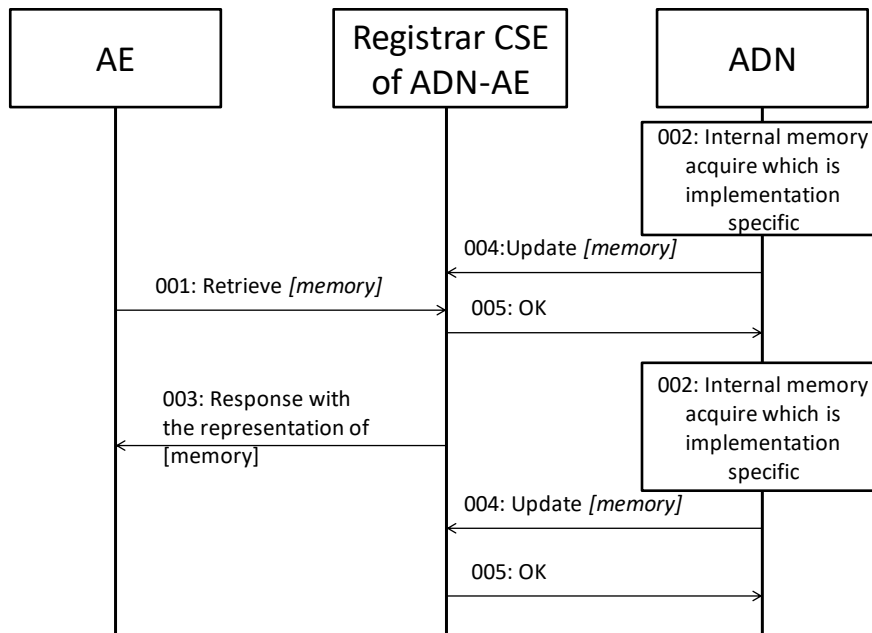


Figure 5.4.2-1: Procedures for management of ADN

001: The AE sends Retrieve request to Registrar CSE of ADN-AE. Requesting the *[memory]* resource.

002: The ADN internally acquires the available memory information.

003: The Registrar CSE responds with the representation of the *[memory]* resource.

004: The ADN-AE updates the *[memory]* resource using Mca reference point.

005: Update successful

NOTE: 001, 003 and 002, 004 don't have a specific order. The ADN is updating the *[memory]* resource periodically. In 003, the Registrar CSE is responding always the current resource.

Subscription and notification could also be used by the AE to monitor the current status of the *[memory]* resource.

The *[memory]* resource in this case is created by ADN-AE.

5.4.3 Management of NoDN

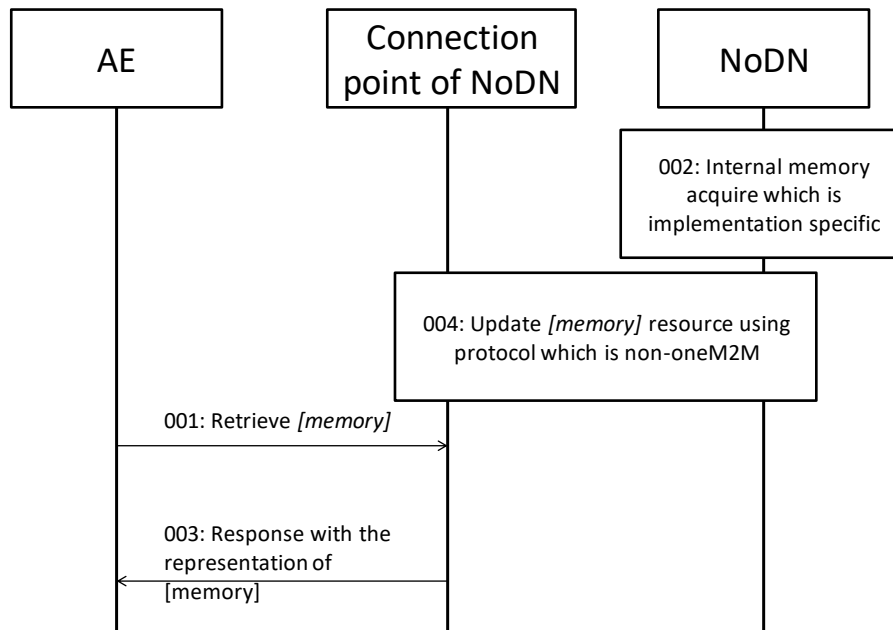


Figure 5.4.3-1: Procedures for management of NoDN

001: The AE sends Retrieve request to connection point of NoDN. Requesting the *[memory]* resource.

002: The NoDN internally acquires the available memory information.

003: The CSE of the connection point responds with the representation of the *[memory]* resource.

004: The NoDN updates the *[memory]* resource using protocol that is non-oneM2M.

NOTE: 001, 003 and 002, 004 don't have a specific order. The NoDN is updating the *[memory]* resource periodically. In 003, the Registrar CSE is responding always the current resource.

The *[memory]* resource in this case is created by NoDN via non oneM2M interface.

5.5 Implementation

5.5.1 Introduction

In this implementation, the current document is using HTTP binding and JSON serialization as examples, and SP-relative-structured resource addressing format is used.

Following resource ID and/or entity ID is used:

- Memory Resource:
 - Resource Name "`memory001`".
 - Resource ID "`m343245334adf`" represented in CSE-relative Unstructured-Resource-ID of the *[memory]* resource.
 - Parent Resource ID "`n893051036jdg`".
- AE-ID "`C324352841DAS`" of the managed entity.
- Registrar/Hosting CSE:
 - SP-relative CSE-ID "`/CSE0034234`".

- CSE Resource Name "server".
- Host name of the Registrar/Hosting CSE "in.management.server.com".

5.5.2 Management of ASN, MN and IN

Table 5.5.2-1

Step	Message Example
001	<pre>HTTP Request GET ~/CSE0034234/server/memory001 HTTP/1.1 Host : in.management.server.com X-M2M-RI : 1234 X-M2M-Origin: C324352841DAS Accept : application/json</pre>
002	<pre>HTTP Response 200 OK X-M2M-RI : 1234 X-M2M-RSC : 2000 { "m2m:mgo" : { "ri": "m343245334adf", "pi": "n893051036jdg", "ct": "20161129T121550", "et": "20181231T235959", "lt": "20161130T152341", "ty": 13, "rn": "memory001", "mgd": 1003, "mma": 800, "mmt": 1024 } }</pre>

5.5.3 Management of ADN

Table 5.5.3-1

Step	Message Example
001	Same as 001 in 5.5.2
002	Same as 002 in 5.5.2
004	<pre>HTTP Request PUT ~/CSE0034234/server/memory001?rcn=0 HTTP/1.1 Host : in.management.server.com X-M2M-RI : 1235 Content-Type: application/json X-M2M-Origin: C324352841DAS { "m2m:mgo": { "mma": 600 } }</pre>
005	<pre>HTTP Response 200 OK X-M2M-RI : 1235 X-M2M-RSC : 2004</pre>

5.5.4 Management of NoDN

001 and 003 is the same as in clause 5.5.2.

002 and 004 is protocol specific and even proprietary, thus it's out of scope of oneM2M.

6 Device management using external management technologies

6.1 Introduction

In the case of device management using external management technologies such as OMA DM, LWM2M or BBF TR-069 [i.3]. All management is done through IN-CSE, since only the IN-CSE has the capability to talk to the Management Server. Therefore, in this case, all <node> resource that represents the managed entity is hosted under the <CSEBase> of IN-CSE. The managed entity may be IN, MN, ASN, ADN or NoDN. Then the <mgmtObj> that is used for management is hosted as the child resource of <node> resource. The attribute *objectIDs* and *objectPaths* of the <mgmtObj> are set in this case to map to the external management objects.

6.2 Use case

An application want to retrieve the current available memory of the device. In this case, the specialization of <mgmtObj> memory is used clause D.4 of [i.2]. In the memory specialization, *memAvailable* is the attribute that contains the required information. The application can get to know the available memory by retrieving the resource.

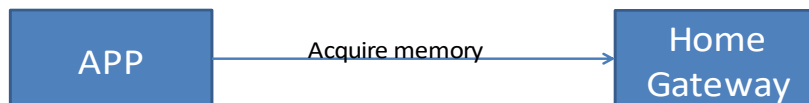


Figure 6.2-1: Use case for management

6.3 Architecture

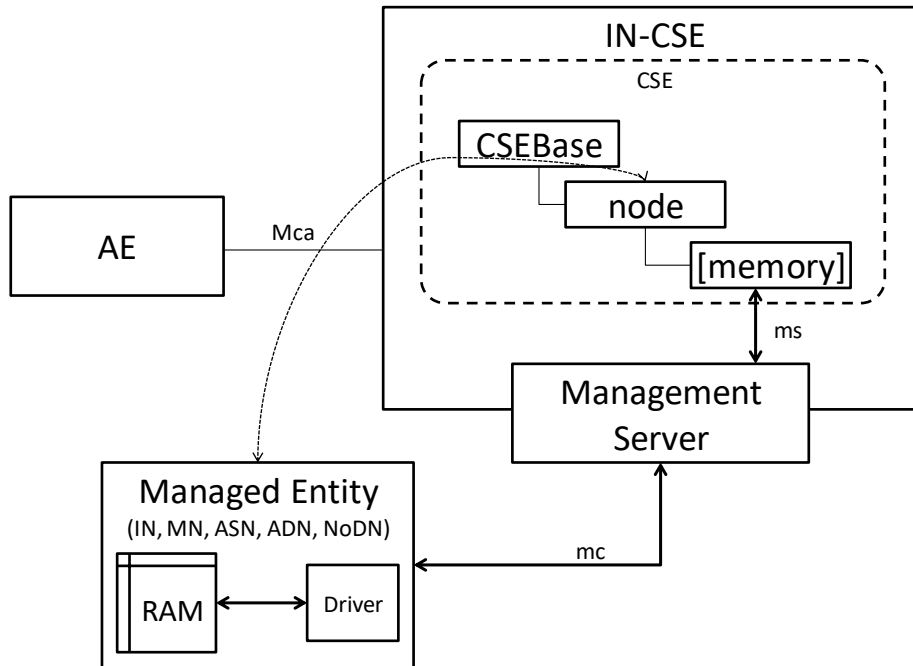


Figure 6.3-1: Architecture for management using external technology

In the case of management using external management technology, all management related resources are hosted on the IN-CSE. IN-CSE interacts with Management Server through *ms* interface and triggers the Management Server to send management commands to Managed Entity through the *mc* interface. *mc* and *ms* interface is defined by external management technology or implementation specific which is out of scope of oneM2M.

The Management Server may be deployed together with the IN-CSE, in this case, the IN-CSE directly sends out management commands through *mc* interface to the Managed Entity. The Management Server may also be deployed separately from the IN-CSE, in this case, the IN-CSE need to send commands to Management Server to trigger tell the Management Server to send out management commands to the Managed Entity. *ms* interface may be implementation specific.

6.4 Procedures

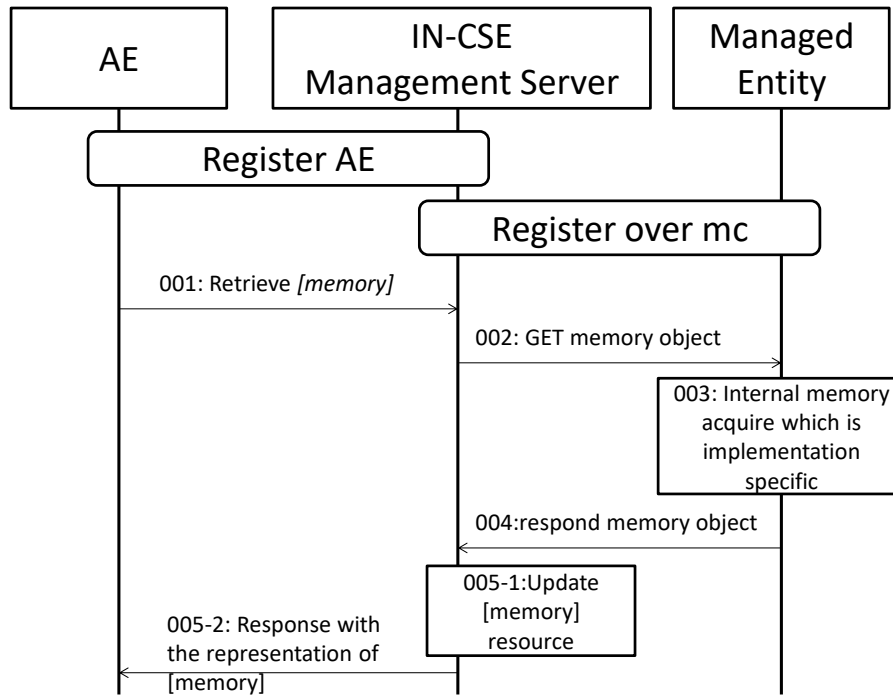


Figure 6.4-1: Procedures for management to ADN

001: AE retrieves *[memory]* resource specialization from IN-CSE over *mca* interface.

002: On receiving the request, the IN-CSE sends management command to managed entity using external management technology implementation.

003: On the managed entity, the driver acquires the memory information.

004: The managed entity responds with acquired memory information over the *mc* interface.

005: The IN-CSE updates the *[memory]* resource and respond the RETRIEVE request with the representation of *[memory]* resource.

6.5 Implementation

In the implementation, the document is using LWM2M as the external management technology. Resource ID and/or entity ID that are defined in clause 5.5.1 is also applicable to clause 6.5.

In addition, host name of the managed entity is named as "*adn.managed.entity.com*", the memory resource name is "*memory002*", the memory resource ID is "*m463245245afg*", and the parent resource ID of the memory resource is "*n993051028asd*". The request URL for the managed entity is assumed as "*coap://adn.managed.entity.com/adn001/memory*".

Table 6.5-1

Step	Message Example
001	<p>HTTP Request</p> <pre>GET ~/CSE0034234/server/memory002 HTTP/1.1 Host : in.management.server.com X-M2M-RI : 3454 X-M2M-Origin: C324352841DAS Accept : application/json</pre>
002	<p>CoAP Request</p> <pre>Method : 0.01 (GET) Uri-Host : adn.managed.entity.com Uri-Path : adn001 Uri-Path : memory oneM2M-FR : C324352841DAS oneM2M-RQI: 4553 Payload : empty</pre>
004	<p>CoAP Response</p> <pre>X-M2M-RSC : 2.05 (OK) oneM2M-RQI : 4553 Content-Format: 50 (application/json) Payload : {memory object serialized in JSON}</pre>
005-2	<p>HTTP Response</p> <pre>X-M2M-RI : 3454 X-M2M-RSC : 2000 { "m2m:mgo" : { "ri": "m463245245afg", "pi": "n993051028asd", "ct": "20161129T121550", "et": "20181231T235959", "lt": "20161130T152341", "ty": 13, "rn": "memory002", "mgd": 1003, "mma": 800, "mmt": 1024 } }</pre>

History

Publication history		
V2.0.0	12-Mar-2018	Release 2A - Publication