ONEM2M
TECHNICAL SPECIFICATION

Document Number	oneM2M-TS-0001 - V-2014-08
Document Name:	oneM2M Functional Architecture Baseline Draft
Date:	2014-08-01
Abstract:	This document specifies the initial relase of the functional architecture for the oneM2M Services Platform.

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

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

This is a draft oneM2M document and should not be relied upon; the final version, if any, will be made available by oneM2M Partners Type 1.

1 Scope

The present document describes the end-to-end oneM2M functional architecture, including the description of the functional entities and associated reference points.

oneM2M functional architecture focuses on the Service Layer aspects and takes Underlying Network-independent view of the end-to-end services. The Underlying Network is used for the transport of data and potentially for other services.

2 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.

2.1 Normative references

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

[1] oneM2M Security Solutions Technical Specification.

2.2 Informative references

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 TS-0002: "oneM2M Requirements Technical Specification".

[i.2] oneM2M TS-0004: "oneM2M Protocol Specification".

[i.3] oneM2M TS-0003: "oneM2M Security Solutions".

[i.4] TR-069: “CPE WAN Management Protocol Issue”: 1 Amendment 5, November 2013, Broadband Forum.

[i.5] OMA-DM: “OMA Device Management Protocol”, Version 1.3, Open Mobile Alliance.

[i.6] LWM2M: "OMA LightweightM2M”, Version 1.0, Open Mobile Alliance.

[i.7] OMA-TS-MLP-V3-4-20130226-C: "Mobile Location Protocol", Version 3.4.

[i.8] OMA-TS-REST-NetAPI_TerminalLocation-V1_0-20130924-A: "RESTful Network API for Terminal Location", Version 1.0.

[i. 9] IETF RFC 1035: "Domain names - Implementation and specification".

[i.10] IETF RFC 3588: "Diameter Base Protocol".

[i.11] IETF RFC 3596: "DNS Extensions to Support IP Version 6".

[i.12] IETF RFC 3986: "Uniform Resource Identifier (URI): General Syntax".

[i.13] IETF RFC 4006: "Diameter Credit-Control Application".

[i.14] IETF RFC 6874: "Representing IPV6 Zone Identifiers in Address Literals and Uniform Resources Identifiers".

[i.15] IETF RFC 6895: "Domain Name System (DNS) IANA Considerations".

[i.16] GSMA-IR.67: "DNS/ENU Guidelines for Service Providers & GRX/IPX Providers".

[i.17] 3GPP TS 23 682: Architecture enhancements to facilitate communications with packet data networks and applications (Release 11)".

[i.18] ETSI TS 132 240: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Telecommunication management; Charging management; Charging architecture and principles (3GPP TS 32.240)".

[i.19] ETSI TS 132 299: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Telecommunication management; Charging management; Diameter charging applications (3GPP TS 32.299)".

[i.20] 3GPP2 X.P0068: "Network Enhancements for Machine to Machine (M2M)".

[i.21] JNI 6.0 API Specification: "Java Native Interface 6.0 Specification".

[i.22] 3GPP TS 23 401: "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access".

[i.23] 3GPP TS 23 402: " Architecture enhancements for non-3GPP accesses".

[i.24] 3GPP TS 23 060: " General Packet Radio Service (GPRS); Service description; Stage 2".

[i.25] 3GPP TS 22 368: "Service requirements for Machine Type Communications (MTC); Stage 1"

[i.26] 3GPP TS 23 003: "Numbering, addressing and identification".

[i.27] Recommendation ITU-T X.660 | ISO/IEC 9834-1: "Information technology - Procedures for the operation of object identifier registration authorities: General procedures and top arcs of the international object identifier tree".

3 Definitions, abbreviations and acronyms

3.1 Definitions

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

application layer: comprises oneM2M Applications and related business and operational logic

attribute: stores information pertaining to the resource. An attribute has a name and a value. Only one attribute with a given name can belong to a given resource. For an attribute defined as having “multiplicity” greater than 1, the value of that attribute is a composite value, i.e. a list of different values.

child resource: sub-resource of another resource that is its parent resource

NOTE: The parent resource contains references to the child resources(s).

common services layer: consists of oneM2M service functions that enable oneM2M Applications (e.g. management, discovery and policy enforcement)

Common Services Function (CSF): informative architectural construct which conceptually groups together a number of sub-functions

NOTE: Those sub-functions are implemented as normative resources and procedures. A set of CSFs is contained in the CSE.

execution environment: logical entity that represents an environment capable of running software modules

hosting CSE: CSE where the addressed resource is hosted

M2M service provider domain: is the part of the M2M System that is associated with a specific M2M Service Provider

managed entity: may be either an M2M Device, M2M Gateway, or a device in the M2M Area Network or the M2M Application Layer or M2M Service Layer software components

Management Proxy: An Entity within the Device Management Architecture, in conjunction with the Management Client, that acts as an intermediary between the Management Server and the Proxy Management Client.

network services layer: provides transport, connectivity and service functions

node: functional entity containing one of the following:

- one or more M2M Applications;
- one CSE and zero or more M2M Applications.

originator: For single-hop case, the originator is the entity that sends a Request. For multi-hop case, the originator is the entity that sends the first Request in a sequence of requests.

NOTE: An originator can either be an AE or a CSE.

Proxy Management Client: An Entity within the Device Management Architecture that provides local management capabilities to a device in an M2M Area Network.

receiver: is the entity that receives the Request.

NOTE: A Receiver can a CSE or can be and AE when notification is requested.

registree: is an AE or CSE that registers with another CSE

registrar CSE: CSE is the CSE where an Application or another CSE has registered

resource: is a uniquely addressable entity in oneM2M architecture.

NOTE: A resource is transferred and manipulated using CRUD operations. A resource can contain child resource(s) and attribute(s), which are also uniquely addressable.

service charging and accounting: set of functionalities within the M2M Service Layer that enable configuration of information collection and charging policies, collection of Charging Records based on the policies, and correlation of Charging Records to users of M2M common services

service charging record: formatted collection of information about a chargeable operation

service layer offline charging: mechanism where charging information does not affect, in real-time, the service rendered

service layer online charging: mechanism where charging information can affect, in real-time, the service rendered, including real time credit control

software package: is an entity that can be deployed on the Execution Environment

NOTE: It can consist of entities such as software modules, configuration files, or other entities.

structured data: is data that either has a structure according to a specified Information Model or are otherwise organized in a defined manner

transit CSE: is any receiver CSE that is not a hosting CSE.

3.2 Abbreviations and Acronyms

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

2G	Second Generation
3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
AAA	Authentication, Authorization, Accounting
AAAA	Authentication, Authorization, Accounting and Auditing
A/AAAA	IPv4/IPv6 DNS records that are used to map hostnames to an IP address
ACA	Accounting Answer
ACR	Accounting Request
ADN	Application Dedicated Node
ADN-AE	AE which resides in the Application Dedicated Node
AE	Application Entity
AE-ID	Application Entity Identifier
AID CSF	Addressing and Identification CSF
Annc	Announced
API	Application Program Interface
AS	Application Server
ASM CSF	Application and Service Layer Management CSF
ASN	Application Service Node
ASN-AE	Application Entity that is registered with the CSE at Application Service Node
ASN-CSE	CSE which resides in the Application Service Node
BBF	BroadBand Forum
CDR	Charging Data Record
CF	Configuration Function
CHF	Charging Function
CM	Conditional Mandatory
CMDH	Communication Management and Delivery Handling
COSEM	Companion Specification for Energy Metering
CRUD	Create Retrieve Update Delete
CRUDN	Create Retrieve Update Delete Notify
CSE	Common Services Entity
CSE-ID	Common Service Entity Identifier
CSE-PoA	CSE Point of Access
CSF	Common Services Function
DCF	Device Configuration Function
DDMF	Device Diagnostics and Monitoring Function

DFMF	Device Firmware Management Function
DHCP	Dynamic Host Configuration Protocol
DIS CSF	Discovery CSF
DM	Device Management
DMF	Diagnostic and Monitoring Function
DMG CSF	Device Management CSF
DMR	Data Management and Repository
DNS	Domain Name Server
DTMF	Device Topology Management Function
EF	Enabler Function
FFS	For Further Study
FQDN	Fully Qualified Domain Name
GMG CSF	Group Management CSF
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSMA	GSM Association (Global System for Mobile Communications Association)
HA/LMA	Home Agent/Local Mobility Agent
HAAA	Home AAA
HLR	Home Location Register
HPLMN	Home Public Land Mobile Network
HSS	Home Subscriber Server
HTTP	HyperText Transfer Protocol
IMEI	International Mobile Equipment Identity
IMS	IP Multimedia System
IMSI	International Mobile Subscriber Identity
IN	Infrastructure Node
IN-AE	Application Entity that is registered with the CSE in the Infrastructure Node
IN-CSE	CSE which resides in the Infrastructure Node
IN-DMG	Infrastructure Node Device ManaGement
IN-DMG-MA	Infrastructure Node Device ManaGement Management Adapter
IP	Internet Protocol
IPE	Interworking Proxy application Entity
ISO	International Organization for Standardization
ITU-T	ITU Telecommunication Standardization Sector
IWF	InterWorking Function
JNI	Java Native Interface
LOC	Location
LOC CSF	Location CSF
LWM2M	Lightweight M2M
M2M	Machine to Machine
M2M-IWF	M2M InterWorking Function
M2M-Sub-ID	M2M service Subscription Identifier
MA	Mandatory Announced
Mca	Reference Point for M2M Communication with AE
Mcc	Reference Point for M2M Communication with CSE
Mcc'	Reference Point for M2M Communication with CSE of different M2M Service Provider
Mch	Reference Point for M2M Communication with external charging server
Mcn	Reference Point for M2M Communication with NSE
MIP	Mobile IP
MN	Middle Node
MN-AE	Application Entity that is registered with the CSE in Middle Node
MN-CSE	CSE which resides in the Middle Node
MSISDN	Mobile Subscriber International Subscriber Directory NumberMTC Machine Type Communications
NA	Not Announced
NAT	Network Address Translation
NSE	Network Service Entity
NSSE CSF	Network Service Exposure, Service Execution and Triggering CSF
OA	Optional Announced
OID	Object Identifier
OMA	Open Mobile Alliance
OMA-DM	Open Mobile Alliance Device Management

OSI	Open Service Initiative or Open Service Interconnection
PDSN	Packet Data Serving Node
PMIP	Proxy Mobile IP
PoA	Point of Access
PPP	Point to Point Protocol
RAM	Random Access Memory
REG	Registration
REG CSF	Registration CSF
RFC	Request for Comments
RO	Read Only
RPC	Remote Procedure Calls
RW	Read Write
SCA	Service Charging and Accounting
SCA CSF	Service Charging and Accounting CSF
SCS	Services Capability Server
SDO	Standards Developing Organization
SEC	Security
SEC CSF	Security CSF
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SMF	Software Monitoring Function
SMS	Short Messaging Service
SP	Service Provider
SP-ID	Service Provider Identifier
SSM	Service Session Management
SSM CSF	Service Session Management CSF
SUB CSF	Subscription and Notification CSF
TBD	To Be Determined
Tsms	Interface between Short Message Entity (SME) and Short Message Service Center (SMS SC)
Tsp	Interface between Service Capability Server (SCS) and Machine Type Communication (MTC) InterWorking Function
UE	User Equipment
UL	UpLink
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Universal Resource Name
WLAN	Wireless Local Area Network
WO	Write Once

4 Conventions

The key words "Shall", "Shall not", "May", "Need not", "Should", "Should not" in this document are to be interpreted as described in the oneM2M Drafting Rules [http://member.onem2m.org/website/Procs.aspx].

To improve readability:

- The information elements of oneM2M Request/Response messages will be referred to as parameters. Parameter abbreviations will be written in bold italic.

- The information elements of resources will be referred to as attributes and child resources. Attributes will be written in italics.

5 Architecture Model

5.1 General Concepts

Figure 5.1-1 depicts the oneM2M Layered Model for supporting end-to-end (E2E) M2M Services. This layered model comprises three layers: Application Layer, Common Services Layer and the Underlying Network Services Layer.

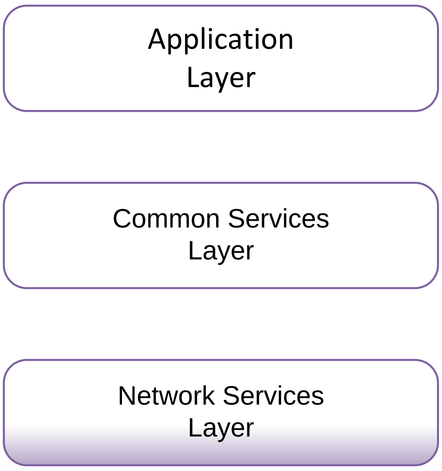


Figure 5.1-1: oneM2M Layered Model

5.2 Architecture Reference Model

5.2.1 Functional Architecture

Figure 5.2.1-1 illustrates the oneM2M functional architecture.

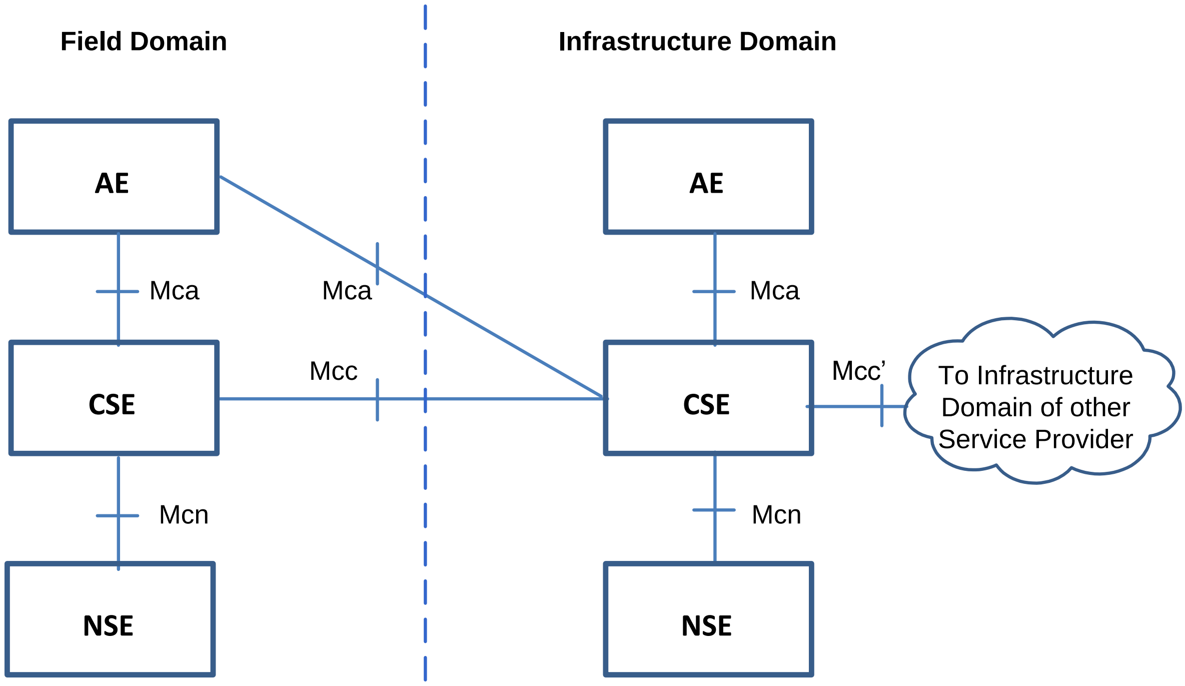


Figure 5.2.1-1: oneM2M Functional Architecture

NOTE 1: Other reference points for charging and management aspects are specified.

- See clause 12.2.1 for Mch reference point.
- See clause 6.2.4 for Mc, Mp, Ms and Ia reference points.

The oneM2M functional architecture in figure 5.2.1-1 comprises of the following functions:

1) **Application Entity (AE):** Application Entity represents an instantiation of Application logic for end-to-end M2M solutions. Each Application Entity is identified with a unique AE-ID (see clause 7.1.2). Examples of the Application Entities can be an instance of a fleet tracking application, a remote blood sugar monitoring application, a power metering application, or a controlling application.

2) **Common Services Entity (CSE):** A Common Services Entity 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 Mca and Mcc. Reference point Mcn is used for accessing Underlying Network Service Entities. Each Common Service Entity is identified with a unique CSE-ID (see clause 7.1.4).

   Examples of service functions offered by CSE include: Data Management, Device Management, M2M Subscription Management, and Location Services. Such "sub-functions" offered by a CSE may be logically and informatively conceptualized as Common Services Functions (CSFs). The normative Resources which implement the service functions in a CSE can be mandatory of optional.,

3) Underlying **Network Services Entity (NSE):** A Network Services Entity provides services from the underlying network to the CSEs. Examples of such services include device management, location services and device triggering. No particular organization of the NSEs is assumed.

NOTE 2: Underlying networks provide data transport services between entities in the oneM2M System. Such data transport services are not included in the NSE.

5.2.2 Reference Points

The following reference points are supported by the Common Services Entity (CSE). The "Mc(-) nomenclature is based on the mnemonic "M2M communications".

NOTE: Information exchange between two M2M nodes assumes the usage of the transport and connectivity services of the Underlying Network Services Entity, which are considered to be the basic services.

5.2.2.1 Mca Reference Point

Communication flows between an Application Entity (AE) and a Common Services Entity (CSE) cross the Mca reference point. These flows enable the AE to use the services supported by the CSE, and for the CSE to communicate with the AE.

NOTE: The AE and the CSE it invokes could or could not be co-located within the same physical entity.

5.2.2.2 Mcc Reference Point

Communication flows between two Common Services Entities (CSEs) cross the Mcc reference point. These flows enable a CSE to use the services supported by another CSE.

5.2.2.3 Mcn Reference Point

Communication flows between a Common Services Entity (CSE) and the Network Services Entity (NSE) cross the Mcn reference point. These flows enable a CSE to use the supported services (other than transport and connectivity services) provided by the NSE.

5.2.2.4 Mcc' Reference Point

Communication flows between two Common Services Entities (CSEs) in infrastructure nodes that are oneM2M compliant and that resides in different M2M SP domains cross the Mcc' reference point. These flows enable a CSE of an infrastructure node residing in the Infrastructure Domain of an M2M Service Provider to communicate with a CSE of another infrastructure node residing in the Infrastructure Domain of another M2M Service Provider to use its supported services, and vice versa.

Mcc' extends the reachability of services offered over the Mcc reference point, or a subset thereof.

The trigger for these communication flows may be initiated elsewhere in the oneM2M network.

5.2.2.5 Mch Reference Point

Communication flows which transfer CDRs generated by the IN to an external charging server cross the Mch reference point. The Mch reference point may be mapped to reference points of other specifications. E.g. for a 3GPP Underlying Network, the Mch reference point maps to the Rf reference point enabling a 3GPP charging server to be used for oneM2M CDRs.

6 Configurations supported by oneM2M Architecture

6.1 Relationships among oneM2M Entities

Figure 6.1-1 illustrates the relationships among oneM2M entities supported by the oneM2M architecture as shown in figure 5.2.1-1. The illustration does not constrain the multiplicity of the entities nor require that all relationships shown are present.

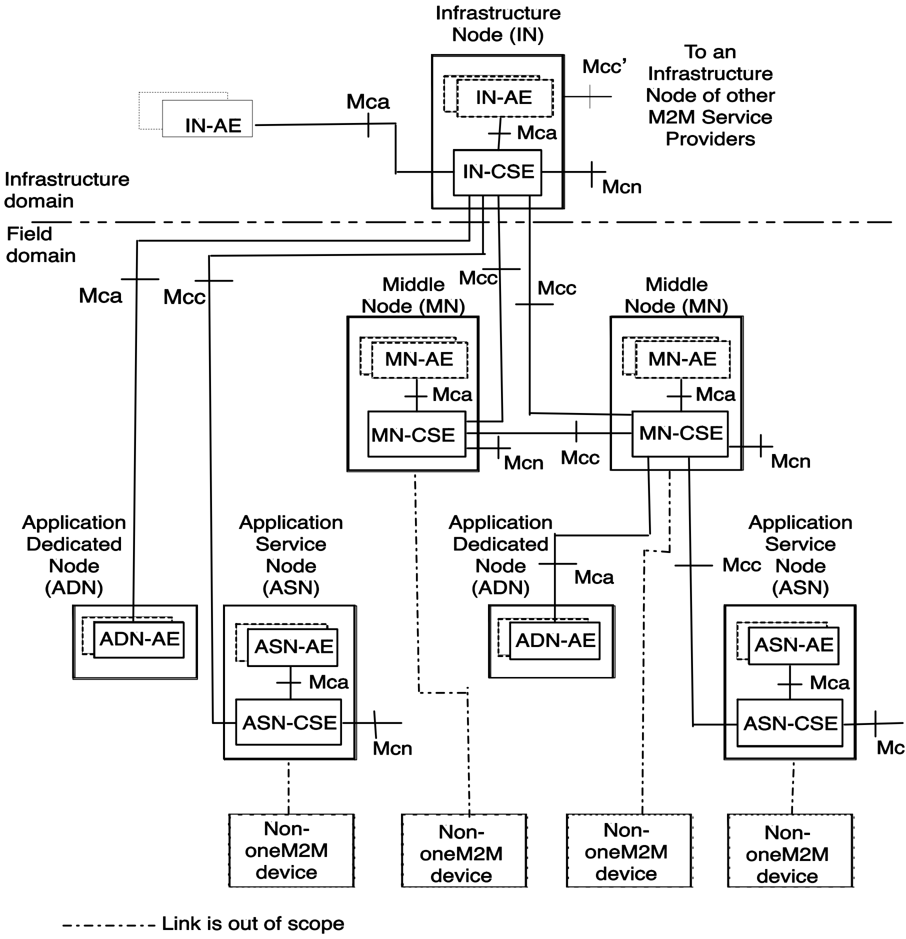


Figure 6.1-1: Configurations supported by oneM2M Architecture

Nodes:

Two types of oneM2M Nodes are supported:

- A CSE-Capable Node is a functional entity that contains one oneM2M Common Services Entity and zero or more oneM2M Application Entities. The ASN and MN are examples of CSE-Capable Nodes.

- A Non-CSE-Capable Node is a functional entity that contains one or more oneM2M Application Entities and no oneM2M Common Services Entity. The ADN is an example of a Non-CSE-Capable Node.

NOTE: CSEs resident in different Nodes could not be identical and are dependent on the services supported by the CSE in that Node.

Description of Nodes:

oneM2M architecture enables the following types of Nodes. As functional objects, such Nodes may or may not be mapped to physical objects.

Application Service Node (ASN):

An Application Service Node is a Node that contains one Common Services Entity and contains at least one Application Entity. There may be zero or more ASNs in the Field Domain of an M2M System.

An Application Service Node may communicate over a Mcc reference point with either:

- exactly one Middle Node;
- or exactly one Infrastructure Node.

An Application Service Node communicates over Mcn with NSEs.

Example of physical mapping: an Application Service Node could reside in an M2M Device.

Application Dedicated Node (ADN):

An Application Dedicated Node is a Node that contains at least one Application Entity and does not contain a Common Services Entity. There may be zero or more ADNs in the Field Domain of an M2M System.

An Application Dedicated Node communicates with a Middle Node or an Infrastructure Node over a Mca reference point.

Example of physical mapping: an Application Dedicated Node could reside in a constrained M2M Device.

Middle Node (MN):

A Middle Node is a Node that contains one Common Services Entity and contains zero or more Application Entities. There may be zero or more Middle Nodes in the Field Domain of an M2M System.

A Middle Node communicates with:

1) either an IN or another MN over an Mcc reference point; plus at least
2) an IN/MN/ASN over an Mcc reference point; or
3) an ADN over a Mca reference point.

A Middle Node communicates over Mcn with NSEs.

Example of physical mapping: a Middle Node could reside in an M2M Gateway.

Infrastructure Node (IN):

An Infrastructure Node is a Node that contains one Common Services Entity and contains zero or more Application Entities. There is one logical Infrastructure Node in the Infrastructure Domain per oneM2M Service Provider of an M2M System. It may contain CSE functions not applicable to other node types.

An Infrastructure Node communicates over respective Mcc reference points with:

- one or more Middle Node(s); and/or
- one or more Application Service Node(s).

An Infrastructure Node communicates with one or more Application Dedicated Nodes over respective Mca reference points.

An Infrastructure Node communicates over Mcn with NSEs, and over Mcc' with Infrastructure Nodes of other M2M Systems.

Example of physical mapping: an Infrastructure Node could reside in an M2M Service Infrastructure.

6.2 Common Services Functions

This clause describes the services provided by the Common Services Layer in the M2M System. Such services reside within a CSE and are referred to as Common Services Functions (CSFs). The CSFs provide services to the AEs via the Mca reference point and to other CSEs via the Mcc reference point. CSEs interact with the NSE via the Mcn reference point. An instantiation of a CSE in a Node comprises a subset of the CSFs from the CSFs described in the present document.

The CSE descriptions in this clause are provided for the understanding of the oneM2M Architecture functionalities and are informative. The CSFs contained inside the CSE can interact with each other but how these interactions take place are not specified in the present document.

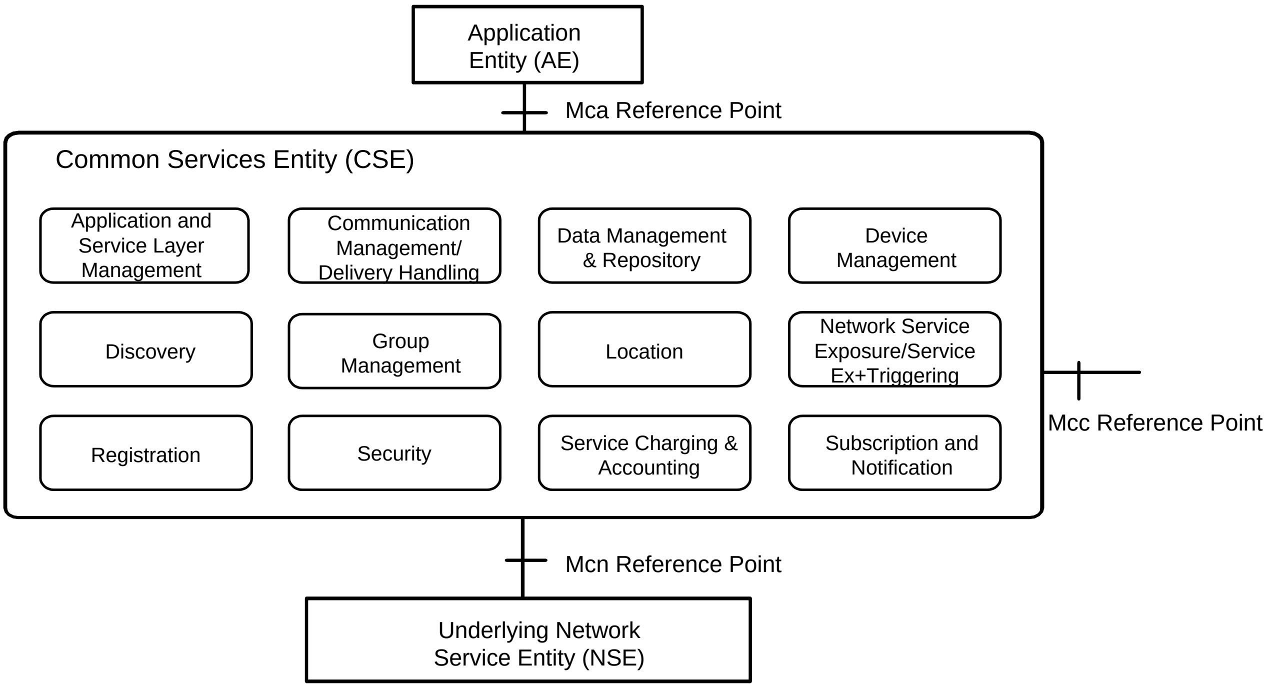

Application Entity (AE)
Mca Reference Point
Common Services Entity (CSE)
Application and Service Layer Management
Communication Management/ Delivery Handling
Data Management & Repository
Device Management
Discovery
Group Management
Location
Network Service Exposure/Service Ex+Triggering
Mcc Reference Point
Registration
Security
Service Charging & Accounting
Subscription and Notification
Mcn Reference Point
Underlying Network Service Entity (NSE)

Figure 6.2-1: Common Services Functions

6.2.1 Application and Service Layer Management

6.2.1.1 General Concepts

The Application and Service Layer Management (ASM) CSF provide management of the AEs and CSEs on the ADNs, ASNs, MNs and INs. This includes capabilities to configure, troubleshoot and upgrade the functions of the CSE, as well as to upgrade the AEs.

6.2.1.2 Detailed Descriptions

The ASM CSF provides management capabilities for CSEs and the AEs.

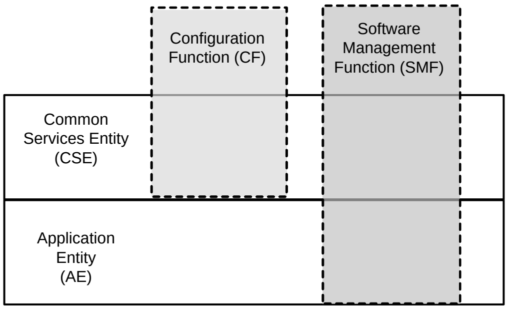

Configuration Function (CF)
Software Management Function (SMF)
Common Services Entity (CSE)
Application Entity (AE)

Figure 6.2.1.2-1: Management Layers and Function

The ASM CSF utilizes the functions provided by the Device Management (DMG) CSF for interaction with the Management Server.

The management functions include:

- Configuration Function (CF): This function enables the configuration of the capabilities and features of the CSE (e.g. CMDH policies).
- Software Management Function (SMF): This function provides lifecycle management for software components and associated artifacts (e.g. configuration files) for different entities such as CSE and AE.

6.2.1.2.1 Software Management Function

The Software Management Function (SMF) provides the capability to manage software components (e.g. Software Package, Software Module) for AEs and CSEs.

The ASM CSF provides the capability to manage the lifecycle of the Software Packages for a CSE or an AE. AE Software Packages may be deployed on any Node that supports the AE; including those on the MNs, ADNs and ASNs.

The lifecycle of a Software Package consists of states (e.g. Installing, Installed, Updating, Uninstalling and Uninstalled) that transition when an action (e.g. Download, Install, Update and Remove) is applied to the Software Package.

When a Software Package is installed into an execution environment the software component that is capable of executing in the Execution Environment is called a Software Module. The lifecycle of a Software Module consists of states (e.g. Idle, Starting, Active, Stopping) that transition when an action (e.g. Start, Stop) is applied to the Software Module.

6.2.2 Communication Management and Delivery Handling

6.2.2.1 General Concepts

The Communication Management and Delivery Handling (CMDH) CSF provides communications with other CSEs, AEs and NSEs.

The CMDH CSF decides at what time to use which communication connection for delivering communications (e.g. CSE-to-CSE communications) and, when needed and allowed, to buffer communication requests so that they can be forwarded at a later time. This processing in the CMDH CSF is carried out per the provisioned CMDH policies and delivery handling parameters that can be specific to each request for communication.

For communication using the Underlying Network data transport services, the Underlying Network can support the equivalent delivery handling functionality. In such case the CMDH CSF uses the Underlying Network, and it may act as a front end to access the Underlying Network equivalent delivery handling functionality.

6.2.2.2 Detailed Descriptions

The service that AEs or CSEs can request from the CMDH CSF is to transport some data to a specific target (CSE or AE), according to given delivery parameters while staying within the constraints of provisioned communication management and delivery handling policies.

The content of the data provided by the Originator is not be visible to the CMDH CSF and does not influence its behaviour. Consequently, the CMDH CSF is not aware of the specific operation requested at the target entity, including the parameters passed to the operation at the destination CSF. This means that all attributes intended to be delivered to the destination entity (e.g. which CSF is the destination on the target entity, what that CSF does with the data etc.) are hidden to the CMDH CSF.

The target entity may be reached either directly or via the CSE(s) of a Middle Node(s).

As part of the delivery request, the CMDH CSF can be provided with acceptable delivery parameters for the Originator (e.g. acceptable expiration time for delivery).

The functions supported by the CMDH CSF are as follows:

- Ability for the M2M Service Provider to derive CMDH policies describing details for the usage of the specific Underlying Network(s). These policies may be based on the M2M Service Subscription associated with Application and Service Entities (AEs and CSEs) in the Field Domain and on the agreements on usage of Underlying Network communication resources. CMDH Policies can be provisioned into the respective CSEs in the Field Domain.

- For the delivery of communication, ability to select appropriate communication path to use at any given time in line with provisioned CMDH policies and with CMDH-related parameters set by the Originator of requests, and when needed and allowed, how long to buffer communication requests so that they can be forwarded at a later time. This policy-driven use of communication resources allows an M2M Service Provider to control which Originators of requests are allowed to consume - possibly costly - communication resources at certain times.

- For the delivery of communication, ability to be aware of the availability of the Underlying Networks.

- Ability to manage the proper use of buffers for store-and-forward processing through use of CMDH policies.

6.2.3 Data Management and Repository

6.2.3.1 General Concepts

One of the purposes of CSEs is to enable AEs to exchange data with each other.

The Data Management and Repository (DMR) CSF is responsible for providing data storage and mediation functions. It includes the capability of collecting data for the purpose of aggregating large amounts of data, converting this data into a specified format, and storing it for analytics and semantic processing. The data can be either raw data transparently retrieved from an M2M Device, or processed data which is calculated and/or aggregated by M2M entities.

NOTE: Collection of large amounts of data is known as the Big Data Repository and is not part of this document.

6.2.3.2 Detailed Descriptions

The DMR CSF provides the capability to store data such as Application data, subscriber information, location information, device information, semantic information, communication status, access permission, etc. The data stored by the DMR CSF enables management of the data and provides the foundation of Big Data.

The following are examples of DMR CSF functionalities:

- Ability to store data in an organized fashion so it is discernible. This includes storage of contextual information such as data types, semantic information, time stamps, location, etc., to complement the data stored in order to access and search the data based on a set of parameters. This is part of data semantics capability which is not part of the present document.

- Provides the means to aggregate data received from different entities.

- Ability to grant access to data from remote CSEs and AEs based on defined access control policies, and trigger data processing based on data access.

- Ability to provide the means to perform data analytics on large amount of data to allow service providers to provide value-added services.

6.2.4 Device Management

6.2.4.1 General Concepts

The Device Management (DMG) CSF provides management of device capabilities on MNs (e.g. M2M Gateways), ASNs and ADNs (e.g. M2M Devices), as well as devices that reside within an M2M Area Network. Application Entities (AE) can manage the device capabilities on those Nodes by using the services provided by the DMG CSF alleviating the need for the AE to have knowledge of the management technology specific protocols or data models. While the AE does not require an understanding of the management technology specific protocols or data models, this information is provided to the AE so that an AE can utilize this information for administrative purposes (e.g. diagnostics, troubleshooting).

6.2.4.1.1 Device Management Architecture

In order to manage the CSE and device capabilities of the MNs, ASNs and ADNs, the DMG can utilize existing device management technologies (e.g. TR-069 [i.4], OMA-DM [i.5], and LWM2M [i.6]) in addition to management of Management Resources across the Mcc reference point. When the device management technology is used to manage the MN, ASN or ADN, the DMG of the IN translates or adapts the management related requests from other CSEs or from AEs to the device management commands of the corresponding device management technology.

In order to perform the translation and adaptation functions, the DMG has a functional component termed the Management Adapter (figure 6.2.4.1.1-1). The Management Adapter in the DMG of the IN (IN-DMG-MA) performs the adaptation between the DMG and Management Servers; while the Management Adapter in the DMG of the MN (MN-DMG-MA) and ASN (ASN-DMG-MA) performs translation and adaptation between the DMG and the Management Client. Only one Management Adapter is shown in the DMG although it can interact with Management Server using different management technologies.

The interface between Management Server and Management Client (figure 6.2.4.1.1-1) is the **mc** interface which is subject to the device management technology that is used (e.g. TR-069 [i.4] or LWM2M [i.6]). The **mc** interface is technology dependent and is outside the scope of this specification.

The DMG in the CSE of the MN has the same functionality as the DMG in the CSE of the ASN. In addition, the DMG in the MN can be used to manage devices in the M2M Area Network. In this case, the DMG is deployed with proxy functionality that interacts with the Proxy Management Client using the **mp** interface. The **mp** interface is technology dependent and is outside the scope of this specification.

The Management Server and Management Client can be implemented as an entity external to the Node or they can be implemented as an entity embedded within the Node (figure 6.2.4.1.1-1). The Management Server and the Management Client are located on the boundary of the Node to indicate this situation as well as to depict that an IN can utilize multiple Management Servers from various M2M and Network Service Providers.

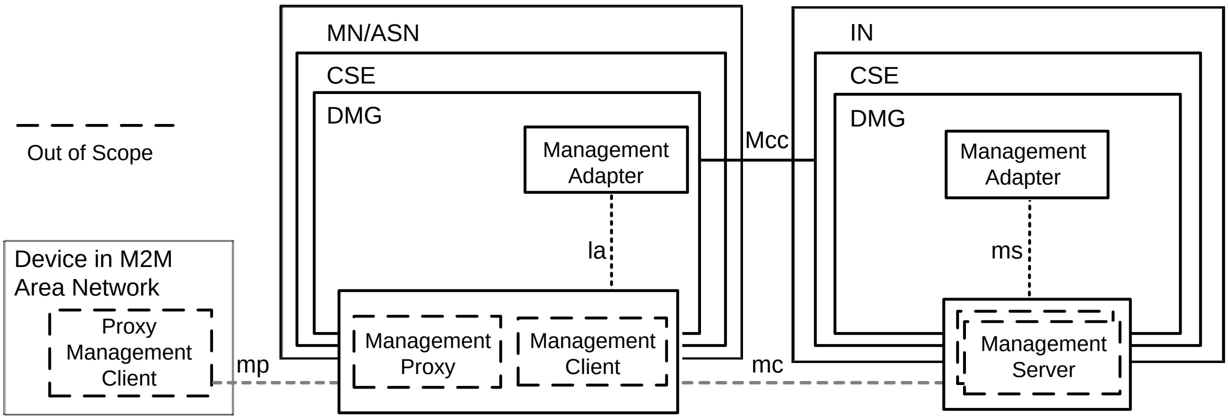


Figure 6.2.4.1.1-1: Device Management Architecture

6.2.4.1.2 Management Server Interaction

6.2.4.1.2.1 Overview

The DMG CSF in the IN has the capability to utilize Management Servers from existing device management technologies (e.g. TR-069 [i.4], OMA DM [i.5], LWM2M [i.6]) to implement the Device Management functions. The IN-DMG-MA communicates with the Management Server using the **ms** interface that is provided by the Management Server. Note that **ms** interface is outside the scope of this specification. The IN-DMG-MA takes the following roles:

- Protocol Translation between DMG and the Management Server:
  - After the DMG receives the requests from the request Originator, the IN-DMG-MA translates the requests from the request Originator to requests with associated identifiers that can be understood by the Management Server. Likewise the IN-DMG-MA translates events from the Management Server and delivers the events to M2M Entities (e.g. AE, CSE) that are subscribed to the event. When the Management Server is embedded within the IN-DMG, the Management Adapter translates the request and accepts events in the protocol understood by the Management Client.
- Interaction with the Management Server:
  - By using **ms** interface, the IN-DMG-MA can communicate with the Management Server. This is for delivering the requests from the request Originator to the Management Server, or receiving information from the Management Server that will be notified to subscribing M2M Entities (e.g. AE, CSE). The communication between the IN-DMG-MA and the Management Server requires an establishment of a session. The establishment of a session between the IN-DMG-MA and Management Server provides security dimensions for Access Control, Authentication, Non-repudiation, Data confidentiality, Communication security, Data integrity and Privacy. The IN-DMG-MA can utilize a policy that defines when a session between the IN-DMG-MA and Management Server is established and torn down.
- Management Server selection:
  - When the IN-DMG-MA communicates with multiple Management Servers that have different level of access control privileges to resources from the Management Server the IN-DMG-MA selects the proper Management Server that has the access control privileges to perform the management requests. The access control policy information for resources from Management Servers may be discovered using the **ms** interface.
- Discovery of external management objects:
  - When the IN-DMG-MA maintains information (i.e. metadata, values) of the external management objects managed by a Management Server using the **ms** interface, the IN-DMG-MA will be capable of discovering and keep up to date the external management object's information that are managed by the IN-DMG and a Management Server.

A Management Server can be located in the Underlying Network using the Mcn reference point as depicted in figure 6.2.4.1.2.1-1 or the Management Server can be located in the M2M Service Layer as depicted in figure 6.2.4.1.2.1-2.

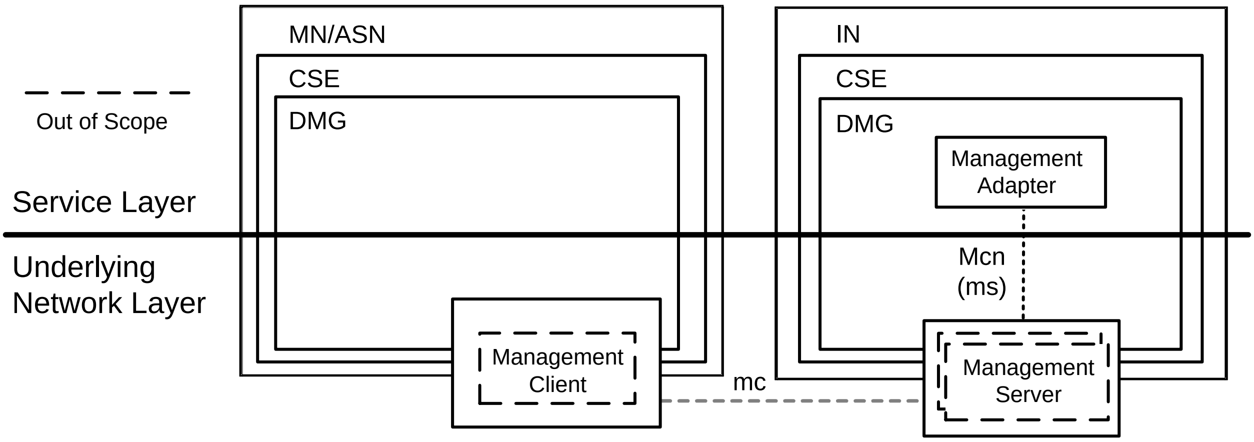


Figure 6.2.4.1.2.1-1: Management Server in Underlying Network

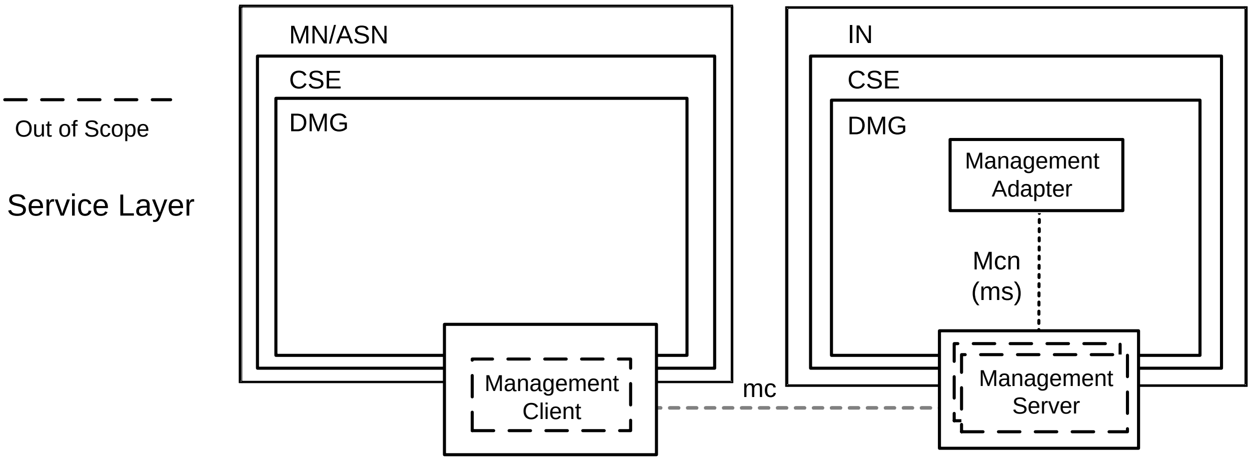


Figure 6.2.4.1.2.1-2: Management Server in M2M Service Layer

The **ms** interface is functionally the same interface regardless if the Management Server resides in the Underlying Network or the Service Layer. However, the access control privileges that the Management Server has for resources from the management technology can be different depending whether the Management Server resides in the Underlying Network or in the Services Layer. For example in figure 6.2.4.1.2.1-1, the Management Server in the Underlying Network controls access of the exposed resources from the management technology while, in the figure 6.2.4.1.2.1-2, the Management Server in the M2M Service Layer controls access to the resources.

6.2.4.1.2.2 Management Server - Access Permissions

When an operation on a M2M Service Layer Resource is performed and if the access to the Resource is granted and the operation for the Resource utilizes a Management Server external to the service layer, the IN-DMG CSF selects one or more among the authenticated Management Servers necessary to access the requested resources. The procedure for the selection of Management Servers is implementation specific and outside the scope of the present document.

The DMG CSF management functions that cause impacts to the Underlying Network utilize access permissions that are delegated from the provider of the network service layer.

6.2.4.1.2.3 Management Server - External management object discovery

An IN-DMG-MA discovers information of the external management objects managed by a Management Server using the **ms** interface. The discovery of this information includes the:

- M2M devices, devices in the M2M Area Network and M2M Applications to which the Management Server has access.

- The metadata associated with the external management objects associated the M2M devices, devices in the M2M Area Network and M2M Applications. This metadata includes items such as the supported data/object model.

The IN-DMG-MA is capable of being kept up-to-date of the changes in the M2M Devices, devices in the M2M Area Network and M2M Applications or the metadata of the external management objects associated with those entities. In addition, the IN-DMG-MA can maintain the value associated external management objects, associated the M2M devices, devices in the M2M Network and M2M Applications.

6.2.4.1.3 Management Client Interaction

6.2.4.1.3.1 Overview

The DMG CSF in the MN or ASN can use the Management Client from existing management technologies (e.g. TR-069 [i.4], OMA DM [i.5], LWM2M [i.6]) to implement the Device Management functions. The MN-DMG-MA or ASN-DMG-MA communicates with the Management Client using the **la** interface (e.g. DM-7, 8, 9 ClientAPI in OMA DM [i.5]) that is provided by the Management Client. Note that the **la** interface is outside the scope of this specification. The MN-DMG-MA or ASN-DMG-MA takes the following roles:

- Interaction with the Management Client:
  - By using **la** interface, the Management Adapter can communicate with the Management Client to discover the external management objects supported by the Management Client.
- Mapping between the DMG and Management Client:
  - After the Management Adapter discovers the external management objects supported by the Management Client; the Management Adapter performs the mapping between the external management objects to resources. The DMG in the MN or ASN can create those resources in the IN-CSE, and the resources can be used by the IN-AE to manage the device capabilities pertaining to the MN or ASN.
  -

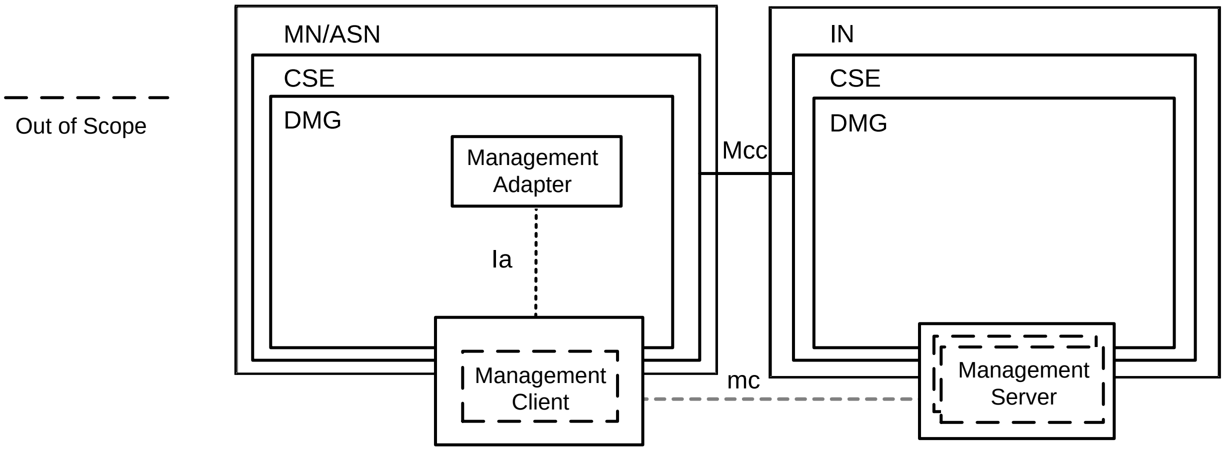


Figure 6.2.4.1.3.1-1: Management Client Interaction using "la" interface

6.2.4.1.4 Device Management Resource Lifecycle

6.2.4.1.4.1 Resource Attributes from Device Management Resources

The lifecycle of a Device Management Resource is implemented using the resource management information defined in clause 9.1 through clause 9.5 and the corresponding procedures to Create, Retrieve, Update and Delete the resources are defined in clause 10. This clause describes additional resource management and procedures for Device Management Resources.

6.2.4.1.4.2 Overview

Clauses 9.1 through 9.5 define resource management information that is applicable to any type of resource, including Device Management Resources. In addition a Device Management Resource also maintains information and relationships that are specific to Device Management Resources. This information is used to:

- Manage external management objects via a Management Server which requires the information necessary to identify and access the Management Server.
- Invoke the security mechanism of the Management Server in order to authorize access to the external management objects.

6.2.4.1.4.3 Procedures for Creation, Update and Deletion of Device Management Resources

Clause 10 defines the procedures to Create, Update and Delete resources. These procedures are also applicable to Device Management Resources in addition to the procedures Device Management Resources are Created, Updated or Deleted:

- By administrative means using the Mca reference point.
- Directly by a CSE based on a discovery or another event within the CSE.
- Indirectly by the Management Server or Management Client when an event (such as firmware update, or fault notification) occurs within the Management Server or Client.

Regardless of the Create, Update or Delete operation, the Originator of the operation will be authorized to perform the operation. In addition, at most one Management Server is able to Create, Delete or Update addressable elements of a Management Resource.

6.2.4.2 Detailed Descriptions

The DMG CSF provides capabilities for the purpose of managing M2M Devices/Gateways as well as devices in M2M Area Networks.

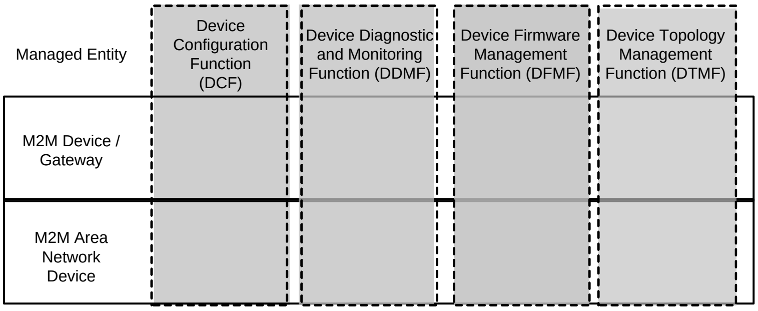

Managed Entity
Device Configuration Function (DCF)
Device Diagnostic and Monitoring Function (DDMF)
Device Firmware Management Function (DFMF)
Device Topology Management Function (DTMF)
M2M Device / Gateway
M2M Area Network Device

Figure 6.2.4.2-1: Device Management Entities and Functions

Such capabilities include:

- Device Configuration Function (DCF): This function includes the configuration of the capabilities of the M2M Device, M2M Gateway or device in the M2M Area Network.
- Device Diagnostics and Monitoring Function (DDMF): This function includes the troubleshooting through the use of diagnostic tests and retrieval of operational status and statistics associated with the M2M Device, M2M Gateway or device in the M2M Area Network.
- Device Firmware Management Function (DFMF): This function provides the software lifecycle management for firmware components and associated artifacts for the M2M Device, M2M Gateway or device in the M2M Area Network.
- Device Topology Management Function (DTMF): This function provides the management of the topology of the M2M Area Network. A M2M Area Network is comprised of ADNs and other devices in the M2M Area Network.

6.2.4.2.1 Device Configuration Function

The Device Configuration Function (DCF) provides the configuration of device capabilities that are necessary to support M2M Services and AEs in M2M Devices, M2M Gateways or devices in a M2M Area Network.

These device configuration capabilities include:

- Discovery of a device's management objects and attributes.
- Ability to enable or disable a device capability.
- Provisioning configuration parameters of a device.

6.2.4.2.2 Device Diagnostics and Monitoring Function

The Device Diagnostics and Monitoring Function (DDMF) permits the troubleshooting of device capabilities that are necessary to support M2M Services and AEs in M2M Devices, M2M Gateways or devices in a M2M Area Network.

These device diagnostic and monitoring capabilities include:

- Configuration of diagnostics and monitoring parameters on the device.
- Retrieval of device information that identifies a device and its model and manufacturer.
- Retrieval of device information for the software and firmware installed on the device.
- Retrieval of information related to a battery within the device.
- Retrieval of information associated with the memory in use by a device.
- Retrieval of the event logs from a device.
- Device reboot diagnostic operation.
- Device factory reset diagnostic operation.

6.2.4.2.3 Device Firmware Management Function

The Device Firmware Management Function (DFMF) provides lifecycle management for firmware associated with a device.

Device firmware is comprised of firmware modules and artefacts (e.g. configuration files) that are maintained on a device. A device can maintain more than one firmware image and the capability to manage individual firmware images. The firmware lifecycle includes actions to download, update or remove a firmware image. In addition firmware could be downloaded and updated within the same action.

6.2.4.2.4 Device Topology Management Function

The Device Topology Management Function (DTMF) is a function that is specific to M2M Gateways where a M2M Gateway maintains zero or more M2M Area Networks.

These device topology management capabilities include:

- Configuration of the topology of the M2M Area Network.
- Retrieval of information related to the devices attached to the M2M Area Network.
- Retrieval of information that describes the transport protocol associated with the M2M Area Network.
- Retrieval of information that describes the characteristics associated with online/offline status of devices in the M2M Area Network.

6.2.5 Discovery

6.2.5.1 General Concepts

The Discovery (DIS) CSF searches information about applications and services as contained in attributes and resources. The result of a discovery request from an Originator depends upon the filter criteria and is subject to access control policy allowed by M2M Service Subscription. An Originator could be an AE or another CSE. The scope of the search could be within one CSE, or in more than one CSE. The discovery results are returned back to the Originator.

6.2.5.2 Detailed Descriptions

The DIS CSF uses the Originator provided filter criteria (e.g. a combination of keywords, identifiers, location and semantic information) that can limit the scope of information returned to the Originator.

The discovery request indicates the address (e.g. URI) of the resource where the discovery is to be performed. Upon receiving such request, the DIS CSF discovers, identifies, and returns the matching information regarding discovered resources according to the filter criteria.

A successful response includes the discovered information or address(es) (e.g. URI(s)) pertaining to the discovered resources. In the latter case the Originator can retrieve the resources using such discovered address. Based on the policies or Originator request, the CSE which received the discovery request can forward the request to other registered ASN-CSEs, MN-CSEs or IN-CSEs.

6.2.6 Group Management

6.2.6.1 General Concepts

The Group Management (GMG) CSF is responsible for handling group related requests. The request is sent to manage a group and its membership as well as for the bulk operations supported by the group. When adding or removing members to/from a group, it is necessary to validate whether the group member complies with the purpose of the group. Bulk operations include read, write, subscribe, notify, device management, etc. Whenever a request or a subscription is made via the group, the group is responsible for aggregating its responses and notifications. The members of a group can have the same role with regards to access control policy control towards a resource. In this case, access control is facilitated by grouping. When the Underlying Network provides broadcasting and multicasting capability, the GMG CSF is able to utilize such capability.

6.2.6.2 Detailed Descriptions

The GMG CSF enables the M2M System to perform bulk operations on multiple devices, applications or resources that are part of a group. In addition, the GMG CSF supports bulk operations to multiple resources of interest and aggregates the results. It facilitates access control based on grouping. When needed and available, the GMG CSF can leverage the existing capabilities of the Underlying Network including broadcasting/multicasting.

When facilitating access control using a group, only members with the same access control policy towards a resource are included in the same group. Also, only AEs or CSEs which have a common role with regards to access control policy are included in the same group. This is used as a representation of the role when facilitating role based access control.

The service functions supported by the GMG CSF are as follows:

- Handles the requests to create, query, update, and delete a group. An AE or a CSE may request the creation/retrieve/update/deletion of a group as well as the addition and deletion of members of the group.

- Creates one or more groups in CSEs in any of the Nodes in oneM2M System for a particular purpose (e.g. facilitation of access control, device management, fan-out common operations to a group of devices, etc.).

- Handles the requests to retrieve the information (e.g. URI, metadata, etc.) of a group and its associated members.

- Manages group membership and handles requests to add or remove members to and from a group's member list. A member may belong to one or more groups. A group may be a member of another group. When new members are added to a group, the GMG CSF validates if the member complies with the purpose of the group.

- Leverages the capabilities of other CSFs in order to fulfill the functionalities supported by the GMG CSF service functions. Examples include: Security CSF for authentication and authorization.

- Forwards requests to all members in the group. In case the group contains another group as a member, the forwarding process is done recursively, i.e. the nested group forwards the request to its members. After forwarding the request to all members in the group, the GMG CSF generates an aggregated response by aggregating the corresponding responses from the Group members.

- Supports subscriptions to individual groups. Subscriptions to a group is made only if the subscriber is interested in all members of the group. If subscription to a group is made, the GMG CSF aggregates the notifications from the group members, and notifies the subscriber with the aggregated notification. Responses and event notifications relevant to a subscription may be selectively filtered by filtering criteria.

6.2.7 Location

6.2.7.1 General Concepts

The Location (LOC) CSF allows AEs to obtain geographical location information of Nodes (e.g., ASN, MN) for location-based services. Such location information requests can be from an AE residing on either a local Node or a remote Node.

NOTE: Geographical location information can include more than simply the longitude and the latitude information.

6.2.7.2 Detailed Descriptions

The LOC CSF obtains and manages geographical location information based on requests from AEs residing on either a local Node or a remote Node. The LOC CSF interacts with any of the following:

- A location server in the Underlying Network;

- A GPS module in an M2M device; or

- Information for inferring location stored in other Nodes.

In order to update the location information, an AE can configure an attribute (e.g. update period). Based on such defined attributes, the LOC CSF can update the location information using one of the location retrieval mechanisms listed above.

NOTE: The location technology (e.g., Cell-ID, assisted-GPS, and fingerprint) used by the Underlying Network depends on its capabilities.

The functions supported by the LOC CSF are as follows:

- Requests other Nodes to share and report their own or other Nodes' geographical location information with the requesting AEs.

- Provides means for protecting the confidentiality of geographical location information.

6.2.8 Network Service Exposure, Service Execution and Triggering

6.2.8.1 General Concepts

The Network Service Exposure, Service Execution and Triggering (NSSE) CSF manages communications with the Underlying Networks for accessing network service functions over the Mcn reference point. The NSSE CSF uses the available/supported methods for service "requests" on behalf of AEs. The NSSE CSF shields other CSFs and AEs from the specific technologies and mechanisms supported by the Underlying Networks.

NOTE: The NSSE CSF provides adaptation for different sets of network service functions supported by various Underlying Networks.

The network service functions provided by the Underlying Network include service functions such as, but not limited to, device triggering, small data transmission, location notification, policy rules setting, location queries, IMS services, device management. Such services do not include the general transport services.

6.2.8.2 Detailed Descriptions

The NSSE CSF manages communication with the Underlying Networks for obtaining network service functions on behalf of other CSFs, remote CSEs or AEs. The NSSE CSF uses the Mcn reference point for communicating with the Underlying Networks.

The M2M System allows the Underlying Networks to control network service procedures and information exchange over the Underlying Networks while providing such network services. For example, for the 3GPP networks, the Underlying Network can choose to provide the network services based on control plane signalling mechanisms.

Other CSFs in a CSE that need to use the services offered by the Underlying Network use the NSSE CSF.

The service functions supported by the NSSE CSF are as follows:

- The NSSE CSF shields other CSFs and AEs from the specific technology and mechanisms supported by the Underlying Networks.

NOTE: The NSSE CSF provides adaptation for different sets of network service functions supported by various Underlying Networks.

- The NSSE CSF maintains the necessary connections and/or sessions over the Mcn reference point, between the CSE and the Underlying Network when local CSFs are in need of a network service.
- The NSSE CSF provides information to the CMDH CSF related to the Underlying Network so the CMDH CSF can include that information to determine proper communication handling.

6.2.9 Registration

6.2.9.1 General Concepts

The Registration (REG) CSF processes a request from an AE or another CSE to register with a Registrar CSE in order to allow the registered entities to use the services offered by the Registrar CSE.

6.2.9.2 Detailed Descriptions

Registration is the process of delivering AE or CSE information to another CSE in order to use M2M Services.

An AE on an ASN, an MN or an IN performs registration locally with the corresponding CSE in order to use M2M services offered by that CSE. An AE on an ADN performs registration with the CSE on an MN or an IN in order to use M2M services offered by that CSE. An IN-AE performs registration with the corresponding CSE on an IN in order to use M2M services offered by that IN CSE. An AE can have interactions with its Registrar CSE (when it is the target CSE) without the need to have the Registrar CSE register with other CSEs.

The CSE on an ASN performs registration with the CSE in the MN in order to be able to use M2M Services offered by the CSE in the MN. As a result of successful ASN-CSE registration with the MN-CSE, the CSEs on the ASN and the MN establish a relationship allowing them to exchange information.

The CSE on an MN performs registration with the CSE of another MN in order to be able to use M2M Services offered by the CSE in the other MN. As a result of successful MN-CSE registration with the other MN-CSE, the CSEs on the MNs establish a relationship allowing them to exchange information.

The CSE on an ASN or on an MN perform registration with the CSE in the IN in order to be able to use M2M Services offered by the CSE in the IN. As a result of successful ASN/MN registration with the IN-CSE, the CSEs on ASN/MN and IN establish a relationship allowing them to exchange information.

Following a successful registration of an AE to a CSE, the AE is able to access, assuming access privilege is granted, the resources in all the CSEs that are potential targets of request from the Registrar CSE.

The capabilities supported by the REG CSF are as follows:

- Ability for AEs to register to their associated CSE, as per table 6.2.9.2-1;
- Ability for CSE to register to the other CSE, as per table 6.2.9.2-1;
- Ability for an ASN-CSE/MN-CSE to register association of its M2M-Ext-ID (if available) with its CSE-ID, (see clause 7.1.8).
- Ability for an ASN-CSE/MN-CSE to register association of its Trigger-Recipient-ID (if available) with its CSE-ID, (see clause 7.1.8). When Trigger-Recipient-ID is not present, it is assumed that the CSE is not able to receive triggers.

NOTE: Such registrations are applicable to a single M2M Service Provider Domain.

Registration information for a Node includes:

- Identifier of the Node.
- Reachability schedules; which are elements of a Node's policy, and specify when messaging can occur between Nodes. Reachability schedules can be used in conjunction with other policy elements. When reachability schedules are not present in a Node then that Node is expected to be always reachable.

6.2.10 Security

6.2.10.1 General Concepts

The Security (SEC) CSF comprises the following functionalities:

- Sensitive data handling;
- Security administration;
- Security association establishment;
- Access control including identification, authentication and authorization;
- Identity management.

Sensitive data handling functionality in the SEC CSF protects the local credentials on which security relies during storage and manipulation. Sensitive data handling functionality performs other sensitive functions such as security algorithms. This functionality is able to support several cryptographically separated security environments.

Security management capabilities are provided by the Security Administration functionality as specified in TS-0003 [1].

NOTE: ASM and DMG CSFs do not include security management capabilities of the SEC CSF.

Security administration functionality enables services such as the following:

- Creation and administration of dedicated security environment supported by Sensitive Data Handling functionality;
- Post-provisioning of a root credential protected by the security environment;
- Provisioning and administration of subscriptions related to M2M Common Services and M2M Application Services.

Security association establishment functionality establishes security association between corresponding M2M Nodes, in order to provide services such as confidentiality and integrity.

Access control functionality authorizes services and specific operations (e.g. Read/Update) on resources identified and authenticated entities, according to provisioned access control policies and assigned roles.

While unique identifier of an entity are used for authentication and identity management, this functionality provides pseudonyms which serve as temporary identifiers which cannot be linked to the true identity of either the associated entity or its user.

6.2.10.2 Detailed Descriptions

The functionalities supported by the SEC CSF are as follows:

- Sensitive data handling:
  - Provides the capability to protect the local credentials on which security relies during storage and manipulation.
  - Extends sensitive data handling functionality to other sensitive data used in the M2M Systems such as subscription related information, access control policies and personal data pertaining to individuals.
  - Performs other sensitive functions as well, such as security algorithms running in cryptographically separated secure environments.
- Security administration:
  - Creates and administers dedicated secure environment supported by sensitive data handling functionality.
  - Post-provisions master credentials protected by the secure environment.

NOTE: The secure environment can also be pre-provisioned with a master credentials prior to deployment; therefore this capability is not always required. Post-provisioning is required when secure remote provisioning needs to be performed or re-initiated after deployment.

- Provisioning and administration of subscriptions related to M2M Services and M2M application services. Besides the associated master credentials, a subscription includes other information classified as sensitive data such as authorization roles and identifiers for access control management.
- Security association establishment:
  - Establishes security associations between corresponding M2M Nodes in order to provide specific security services (e.g. confidentiality, integrity, or support for application level signature generation and verification) involving specified security algorithms and sensitive data. This involves key derivation based on provisioned master credentials. This functionality of the SEC CSF is mandatory when security is supported.
- Access control:
  - Authorizes services and specific operations (e.g. Read/Update) on resources to identified and authenticated entities, according to provisioned access control policies and assigned roles. This functionality is mandatory when any services relying on authorization and access control are present. Among other usages, the services of this functionality may be applied to personal information as a means to preserve privacy.
- Identity protection:
  - Provides pseudonyms to be used instead of the unique identifiers of an entity to serve as temporary identifiers not linkable to the true identity of either the associated entity or its user.

Detailed functionalities are described in the Security Solutions Technical Specification [1].

6.2.11 Service Charging and Accounting

6.2.11.1 General Concepts

The Service Charging and Accounting (SCA) CSF provides charging functions for the Service Layer. It supports different charging models which also include online real time credit control. The SCA CSF manages service layer charging policies and configuration capturing service layer chargeable events, generating charging records and charging information. The SCA CSF can interact with the charging System in the Underlying Network also. The SCA CSF in the IN-CSE handles the charging information.

6.2.11.2 Detailed Descriptions

The SCA CSF performs information recording corresponding to a chargeable event based on the configured charging policies. The SCA CSF sends the charging information transformed from the specific recorded information to the billing domain by the use of a standard or proprietary interface for charging purposes.

The SCA CSF supports "independent service layer charging" and "correlated charging with the Underlying Network" charging system. For independent service layer charging, only charging functions in the M2M service layer are involved. For correlated charging, charging functions in both the service layer and the Underlying Network are involved.

The SCA CSF supports one or multiple charging models, such as the following:

- Subscription based charging: A service subscriber is charged based on service layer subscriptions.
- Event based charging: Charging is based on service layer chargeable events. A chargeable event refers to the discrete transactions. For example, an operation on data (Create, Update, Retrieve) can be an event. Chargeable event can also be timer based. Chargeable events are configurable to initiate information recording. More than one chargeable event can be simultaneously configured and triggered for information recording.

The Service Layer charging system consists of the following logical functions:

- Charging management function: This function handles charging related policies, configurations, inter-node charging <inter-node charging has not been specified - needs review> function communications and interacting with the charging system in the Underlying Network. Charging related policies.
- Charging triggering function: This function resides in the service layer. It captures the chargeable event and generates recorded information for charging. Recorded information may contain mandatory and optional elements.
- Offline charging function: This function handles offline charging related operations. Offline charging does not affect services provided in real time. Charging triggering information is generated at the CSFs where the chargeable transaction happens. The offline charging function generates service charging records based on recorded information. A service charging record is a formatted collection of information about a chargeable event (e.g. amount of data transferred) for use in billing and accounting.

NOTE: Charging triggering and offline charging function are based on charging policies. The system may record information for other purposes such as for event logging. Some of such information may be applicable for charging purposes.

6.2.12 Subscription and Notification

6.2.12.1 General Concepts

The Subscription and Notification (SUB) CSF provides notifications pertaining to a subscription that tracks changes on a resource (e.g. deletion of a resource). A subscription to a resource is initiated by an AE or a CSE, and is granted by the hosting CSE subject to access control policies. During an active resource subscription, the hosting CSE sends a notification per modification of the resource to the address(es) where the resource subscriber wants to receive it.

6.2.12.2 Detailed Descriptions

The SUB CSF manages subscriptions to resources, subject to access control policies, and sends corresponding notifications to the address(es) where the resource subscribers want to receive them. An AE or a CSE is the subscription resource subscriber. AEs and CSEs subscribe to resources other CSEs. A subscription hosting CSE sends notifications to the address(es) specified by the resource subscriber for modifications on a resource. The scope of a resource subscription includes tracking changes of attribute(s) and child resource(s) of the subscribed-to resource. It does not include tracking the change of attribute(s) of the child resource(s). Filter criteria conditions can be used to constrain the scope of tracking.

A subscription is represented as resource subscription in the CSE resource structure.

The functions supported by the SUB CSF are as follows:

- Inclusion of the resource subscriber ID, the hosting CSE-ID and subscribed-to resource address(es) per resource subscription request. It may also include other criteria (e.g. resource modifications of interest and notification policy) and the address(es) where to send the notifications.
- Ability to subscribe to a single resource via a single subscription, or subscribe to multiple resources via a single subscription when they are grouped and represented as a single group resource.

6.3 Security Aspects

The oneM2M Security Analysis Technical Report [i.3] differentiates security domains related to the transport layer (Underlying Network), service layer (M2M common services) and Application Layer. It also considers possible trust scenarios involving these different security domains, and investigates countermeasures to threats that potentially affect the security of the M2M System.

Each of the security domains may provide their own set of security capabilities. The oneM2M security solution shall provide configurable security services through an API for upper security domains to leverage, or enable the use of the exposed security features of other security domains when appropriate.

As a result, beyond providing security solutions that protect the integrity of the M2M Service Layer, the oneM2M architecture exposes, through its APIs, further security services that are made available to M2M Applications. This enables M2M Applications to benefit from security solutions deployed in the M2M Service Architecture, without adding redundant and/or proprietary security solutions.

NOTE: It remains the responsibility of M2M Application Service Providers to perform their own risk assessment process to identifying the specific threats affecting them and derive their actual security needs.

Security aspects are described in oneM2M Security Solutions Technical Specification [1].

6.4 Intra-M2M SP Communication

A CSE shall perform registration with another CSE to be able to use M2M Services offered by that CSE and to allow the other CSE to use its services. As a result of successful registration the CSEs establish a relationship allowing them to exchange information.

An AE shall perform registration with a CSE in order to be able to use M2M Services offered by that CSE. As a result of successful AE registration, the AE and the CSE establish a relationship allowing them to exchange information.

The following table shows which oneM2M entity types shall be able to register with which other entity types:

Table 6.4-1: Entity Registration

Originator (Registree)	Receiver (Registrar)	Registration Procedure
ADN-AE	MN-CSE, IN-CSE	AE registration procedure see clause 10.1.1.2.2
ASN-AE	ASN-CSE	
MN-AE	MN-CSE	
IN-AE	IN-CSE	
ASN-CSE	MN-CSE, IN-CSE	CSE registration procedure see clause 10.1.1.2.1
MN-CSE	MN-CSE, IN-CSE	

The Originator (Registree) in table 6.4-1 requests the registration and the Receiver (Registrar) is responsible for verifying the request, and checking the authentication and authorization of the Originator in order to establish a peer relationship.

- An AE shall not be registered to more than one CSE (ASN-CSE, MN-CSE or IN-CSE).
- An ASN-CSE shall be able to be registered to at most one other CSE (MN-CSE or IN-CSE).

- An MN-CSE shall be able to be registered to at most one other CSE (MN-CSE or IN-CSE).

An MN-CSE shall be able to support only a single registration towards another MN-CSE or an IN-CSE. A concatenation (registration chain) of multiple uni-directional registrations shall not form a loop. E.g., two MN-CSEs A and B, cannot register with each other. Three MN-CSEs A, B and C, where A registers to B, and B registers to C, then C cannot register to A.

6.5 Inter-M2M SP Communication

6.5.1 Inter M2M SP Communication for oneM2M Compliant Nodes

To enable M2M entities (e.g. CSE, AE) in different M2M Service Provider (SP) domains to communicate, configuration within the M2M domain determines if such a communication is allowed. If allowed, the M2M System shall support routing of the traffic across the originating M2M SP domain and within the target M2M SP domain.

Communication between different M2M SPs which occurs over the reference point Mcc', is subject to business agreements. The offered functionality is typically a subset of the functionality offered over the Mcc reference point.

Any interM2M SP communication in support of a request originating from one M2M SP domain shall be processed and forwarded through the Infrastructure Node of the originating M2M domain towards the Infrastructure Node of the target M2M SP domain and finally forwarded to its target CSE, if different from the Infrastructure Node. Hence the Infrastructure Node in both M2M domains shall be the exit and entry points, respectively, for all inter M2M SP communication traffic.

In this configuration approach, public DNS shall be used to support traffic routing for inter M2M SP communication in accordance with [i.16]. This relies on public domain names being allocated to communicating CSE entities within the oneM2M architecture, and to whom access across domains is permitted through policies. To that effect, an M2M SP supporting inter- M2M SP communication shall ensure that the public domain names for the CSEs whose functionality is available across domains are held in its public DNS and shall always point to the IP address associated with the Infrastructure Node for the domain (being the entry point) for accessibility purposes.

The M2M SP could optionally also have additional policies (example: black list or white list) that governs accessibility from other domains to CSE functionality located within its own domain. These policies are however out of scope of this document.

The public domain names of CSEs to whom access from other domains is allowed by policies, shall be created in the DNS of the M2M SP by the Infrastructure Node at registration time of these CSEs, and shall be removed at de-registration. DNS entries for CSEs can also be created/removed for registered CSEs at any time by the M2M SP through administrative means to handle dynamic policies.

6.5.1.1 Public Domain Names and CSEs

To enable the usage of public DNSs as described above, there is a need for a naming convention for public names for CSEs. This naming convention facilitates the creation of the necessary entries of the public domain names of CSEs in the DNS by the infrastructure node.

CSEs public domain names shall be a sub-domain of the Infrastructure Node's public domain name. This naming convention allows the Infrastructure Node to include the needed DNS entry corresponding to the CSE to whom access from other domains is allowed. This would typically occur when the CSE registers with the Infrastructure Node, subject to policies, or administratively.

Accordingly, the structure of the public domain of the CSEs in IN/MN/ASN shall follow the following naming convention, which relies on the CSE identifier (CSE-ID) as part of the naming convention to facilitate the DNS entry creation:

- Infrastructure Node CSE public domain name: <Infrastructure Node CSE Identifier>.<M2M Service Provider domain name>.

- Middle Node CSE public domain name: <Middle Node CSE Identifier>.<Infrastructure Node public domain name>.

- Application Service Node CSE public domain name: <Application Service Node CSE Identifier>.<Infrastructure Node public domain name>.

Both the MN-CSE and the ASN-CSE public domain names are sub-domains of the Infrastructure Node public domain name.

The A/AAAA records in the DNS, as per [i. 9], [i.11] and [i.15] shall consist of the public domain name of the CSE and the IP address of the M2M Infrastructure Node, since the M2M Infrastructure Node is the entry point of the M2M Service Provider domain name where it belongs to.

Note that entries in the public domain names of the three nodes depicted above do not imply that the actual CSE-Identifier allocated for that node has to be used in the DNS entry. Rather any name, including indeed the CSE Identifier for the node, can be used there as long as the entry resolves to the intended Node.

EXAMPLE:

These 3 host entries are valid entries in the DNS:

- MN-CSEID.IN-CSEID.m2m.myoperator.org
- node1.node2.m2m.myoperator.org
- MN-CSEID.node22.m2m.myoperator.org

6.5.2 Inter M2M SP Generic Procedures

This clause describes the behaviour of the M2M Nodes in support of inter-M2M SP procedures.

6.5.2.1 Actions of the Originating M2M Node in the Originating Domain

The originator in the originating domain can be any M2M node such as ADN, an MN, or an ASN, and shall send a request to the Registrar CSE to retrieve a resource located in another M2M SP domain.

The originator shall use any of the options defined in clause 9.3.1 to identify the target host and resource for that purpose.

6.5.2.2 Actions of the Receiving CSE in the Originating Domain

The receiving CSE in the originating domain shall check if the addressed resource is locally available. If the addressed resource is not locally available, then the request shall be forwarded to the next hop.

If the receiving CSE is on an IN, it shall check if the addressed resource is locally available within its domain. If the addressed resource is not located within its own domain, then the IN shall perform a DNS lookup by using the target hostname provided in the RETRIEVE request. A successful DNS lookup shall return to the origin IN in the originating domain the IP address of the M2M IN residing in the target M2M SP domain.

Subsequently, the IN in the originating domain shall forward the request to the IN of the target domain.

6.5.2.3 Actions in the IN of the Target Domain

The IN is the entry point of the target M2M SP domain. The IN shall check if the addressed resource is a local resource. If it is not a local resource it shall forward the request to the appropriate CSE, after identifying the hosting CSE within its domain, using the pointOfAccess attribute.

Once the request reaches the target hosting-CSE, the CSE shall apply the access control policies applicable to the request. Consequently, the hosting-CSE shall forward the response for the incoming request following the same path of the incoming request.

6.5.3 DNS Provisioning for Inter-M2M SP Communication

As specified previously, any M2M SP supporting inter-M2M SP communication shall ensure that the public domain names for the CSEs whose functionality is available across domains are held in the M2M SP's DNS and shall always point to the IP address associated with the Infrastructure domain CSE (being the entry point) for accessibility purposes.

This implies that the IN-CSE shall be responsible for creating the appropriate entry in the DNS for a successfully registered CSE in the IN-CSE, if the M2M SP policies do allow access to the CSE across multiple M2M domains. Similarly the IN-CSE shall be responsible for deleting the appropriate entry in the M2M SP's DNS for a successfully de-registered CSE in the IN-CSE if the M2M SP policies do allow access to the CSE across multiple M2M domains.

6.5.3.1 Inter-M2M SP Communication Access Control Policies

Additional M2M SP policies that further restrict access to CSEs to requests originating from configured M2M SPs only, can complement the DNS entries created by the IN-CSE. These policies are out of scope of the present document.

6.5.4 Conditional Inter-M2M Service Provider CSE Registration

Inter-M2M Service Provider CSE registration shall be supported to enable M2M entities (e.g., CSE, AE) in peer M2M Service Provider (SP) domains with the ability to create and operate resources with the equivalent set of possibilities as offered in the intra-M2M Service Provider domain, subject to the following:

- The AE or CSE in either domain requires a representation of its own domain, notably the IN-CSE of its domain, in the peer domain to create resources in the peer domain. As an example, when it is required for an AE or a CSE to create and operate under the representation of an IN-CSE resource from a different M2M SP Domain. This enables the AE or CSE to have a behavior that is identical in both the intra- and inter-M2M SP cases.

An AE or CSE that does not require to use the remoteCSE representations of the other domain as parent resources, can create resources in the peer domain if it knows the parent of the resource to be created and as such does not require IN to IN registration. Hence creating subscriptions within a peer M2M SP shall not require IN to IN registration between peer domains (but remains subject to inter –M2M SP business agreements, and access control policies).

Registration between M2M SPs occurs over the reference point Mcc', and is subject to business agreements. These agreements can limit the offered functionalities in comparison to those offered over the Mcc reference point.

No additional security is required respect to the basic procedure as described in 6.5.1, 6.5.2 and 6.5.3.

The following table shows which oneM2M entity types can register with which other entity types across the Mcc' reference point.

Table 6.5.4-1: Inter M2M SP Entity Registration

Originator (Registree)	Receiver (Registrar)	Registration Procedure
IN-CSE	IN-CSE	CSE registration procedure. See clause 10.1.1.2.1

An IN-CSE is allowed to register to the IN-CSE of multiple different M2M SP domains in the oneM2M System.

Any inter-M2M SP communications in support of a request originating from one M2M SP domain shall be processed and forwarded through the IN of the originating M2M domain towards the IN of the target M2M SP domain and finally forwarded to its target CSE, if different from the target domain’s IN. Hence the IN in both M2M domains shall be the exit and entry points, respectively, for all inter-M2M SP communication traffic.

6.6 M2M Service Subscription

The M2M Service Subscription defines the technical part of the contract between an M2M Subscriber (typically an M2M Application Service Provider) and an M2M Service Provider. Each M2M Service Subscription shall have a unique identifier, the M2M-Sub-ID, as specified in clause 7.1.11. An M2M Service Subscription establishes a link between one or more AEs; one or more M2M Nodes, one or more roles associated with an M2M Service Subscription as well as subscriber defined groups used for access control policies.

An M2M Service is the marketable service offered to M2M service subscriber (organizations or individuals). Examples of services include device management, location, data exchange, etc.

In each M2M Service, one or multiple M2M Service role(s) shall be defined by the M2M Service Provider. An M2M Service subscription role is defined as a set of privileges pertaining to a resource types which are associated with M2M Service. The M2M Subscriber subscribes as one or multiple roles within the M2M Services, depending upon which role(s) the M2M Subscribers are interested in.

An example of the set of M2M Services and M2M Service subscription roles is provided in annex G.

It shall be possible for an M2M Service Providers to define their own services and roles.

An M2M Service Subscription governs an M2M subscription, the M2M Nodes where AEs can reside and execute the M2M Service roles and subscriber defined groups used for access control policies. An example of the Subscriber defined group is "all node applications within this service subscription with App-ID set to a specific value". Access control policy resource configuration shall comply with the M2M Service Subscription. For authorization of accessing the resource, access control policy is applied and can be based on M2M Service subscription roles.

How to authorize the request operation based on M2M Service Subscription resource are defined in TS-0003 [1].

An M2M Service Subscription shall be used for the following purposes:

- Serve as a basis for authorization for resource operations.
- Serve as the basis for charging.
- Identify which Nodes are part of this M2M Service Subscription.

7 M2M Identification and addressing

7.1 M2M Identifiers

This clause provides a list of identifiers required for the purpose of interworking within the oneM2M architectural model.

An identifier provides unique information or name, tag or label which has a consistent meaning when applied. For example, a CSE-ID for an MN is used for the purpose of CSE identification within an MN. A CSE-ID for an ASN is used for CSE identification within an ASN.

One identifier shall not be assigned to two or more different entities. An identifier is used to reference M2M entities during their lifetime. Such identifiers provide information for association with other identifiers.

It is assumed that the Application Identifier (App-ID) and the CSE Identifier (CSE-ID) have been assigned initially and are known before an M2M System boots up.

7.1.1 M2M Service Provider Identifier (M2M-SP-ID)

An M2M Service Provider shall be uniquely identified by the M2M Service Provider Identifier (M2M-SP-ID). This is a static value assigned to the Service Provider.

7.1.2 Application Entity Identifier (AE-ID)

An Application Entity Identifier (AE-ID) uniquely identifies an AE resident on an M2M Node, or an AE that requests to interact with an M2M Node. An AE-ID shall identify an Application Entity for the purpose of all interactions within the M2M System.

The AE-ID is globally unique and when used internally to a specific M2M SP domain, it is sufficient to be unique within that M2M Service Provider domain. It is extended to become globally unique when used outside the M2M Service Provider boundaries. The IN-CSE shall perform this task of adding or removing identifier portions (identifying the M2M SP) according to clause 7.1.12.

Additionally the AE-ID, when used in the context of a specific CSE where the AE is registered, it is sufficient to be unique within the scope of that specific CSE. It is extended to become M2M Service Provider unique when used outside such specific CSE.

The hosting CSE of the AE shall perform this task of adding or removing the identifier portions according to clause 7.1.12.

7.1.3 Application Identifier (App-ID)

This is equivalent to the application name and is not guaranteed to be globally unique on its own.

This identifier may be supported via a single or multiple registration authorities/entities. The definition and assignment of such an identifier is out of scope of this document.

NOTE: Detailed format and structure of the App-ID is not specified in the present document. Who assigns the App-ID is not specified in the present document.

7.1.4 CSE Identifier (CSE-ID)

A CSE shall be identified by a globally unique identifier, the CSE-ID, when instantiated within an M2M node in the M2M System.

The CSE-ID is globally unique when used internally within a specific M2M SP domain. It is sufficient to be unique within that M2M Service Provider domain. It is extended to become globally unique when used outside the M2M

Service Provider boundaries. The IN-CSE shall perform this task of adding or removing the identifier portions according to clause 7.1.12.

The CSE-ID shall identify the CSE for the purpose of all interactions from/to the CSE within the M2M System.

7.1.5 M2M Node Identifier (M2M-Node-ID)

An M2M node, hosting a CSE and/or Application(s) shall be identified by a globally unique identifier, the M2M-Node-ID.

The M2M System shall allow the M2M Service Provider to set the CSE-ID and the M2M-Node-ID to the same value.

The M2M-Node-ID enables the M2M Service Provider to bind a CSE-ID to a specific M2M node.

Examples of allocating a globally unique M2M-Node-ID include the use of Object Identity (OID) and IMEI. For details on OID, see annex H.

7.1.6 M2M Service Subscription Identifier (M2M-Sub-ID)

The M2M-Sub-ID enables the M2M Service Provider to bind application(s), M2M nodes, CSEs and services identified by service identifiers to a particular M2M Service Subscription.

The M2M Service Subscription Identifier has the following characteristics:

- belongs to the M2M Service Provider;
- identifies the subscription to an M2M Service Provider;
- enables communication with the M2M Service Provider;
- can differ from the M2M Underlying Network Subscription Identifier.

There can be multiple M2M Service Subscription Identifiers per M2M Underlying Network subscription.

7.1.7 M2M Request Identifier (M2M-Request-ID)

The M2M-Request-ID tracks a Request initiated by an AE over the Mca reference point, and by a CSE over the Mcc reference point, if applicable, end to end. It is also included in the Response to the Request over the Mca or Mcc reference points.

To enable an AE to track Requests and corresponding Responses over the Mca reference point, AEs shall include a distinct M2M Request Identifier per request over the Mca Reference point to the CSE for any initiated request.

The CSE shall make such M2M Request Identifier received from the AE globally unique by appending its CSE-ID to it.

If the CSE creates an M2M Request Identifier, then the CSE shall maintain a binding between the M2M Request Identifier received from the AE and the M2M Request Identifier it created in its interactions towards other peer CSEs. The CSE shall include the M2M Request Identifier received from the AE in its Response to the AE. This binding shall be maintained by the CSE until the Request message sequence is completed. Note that the Request initiated by the CSE could be the result of an application Request, or a request initiated autonomously by the CSE to fulfil a service.

In case the receiving CSE is not reachable over the underlying network, the IN-CSE initiates procedure for "waking up" the Node hosting the receiving CSE by using procedures such as device triggering over the Mcn reference point. For Device Triggering, the triggering reference number used to co-relate device triggering response is independent of the M2M Request Identifier. An IN-CSE may use the same value of an M2M-Request-Identifier in an incoming request for the triggering reference number in its interaction with the underlying network.

A CSE receiving a Request from a peer CSE shall include the received M2M Request Identifier in all additional Requests unspanned (i.e. 1:1) it has to generate (including propagation of the incoming Request) and that are associated with the incoming Request, where applicable.

Note that the M2M Request Identifier can be made globally unique by including the CSE-ID in combination with any random number.

7.1.8 M2M External Identifier (M2M-Ext-ID)

The M2M-Ext-ID is used by an M2M Service Provider (M2M SP) when services targeted to a CSE, identified by a CSE-ID, are requested from the Underlying Network.

The M2M External Identifier allows the Underlying Network to identify the M2M Device (e.g., ASN, MN) associated with the CSE-ID. To that effect, the Underlying Network maps the M2M-Ext-ID to the Underlying Network specific Identifier it allocated to the target M2M Device. In addition, the M2M SP shall maintain the association between the CSE-ID, the M2M-Ext-ID and the identity of the Underlying Network.

Both pre-provisioned and dynamic association between the CSE-ID with the M2M-Ext-ID are supported.

NOTE 1: For each CSE-ID, there is only one M2M-Ext-ID for a specific UNetwork-ID. Hence an M2M SP interworking with multiple Underlying Networks has different M2M-Ext-IDs associated with the same CSE-ID, one per Underlying Network and selects the appropriate M2M-Ext-ID for any service request it initiates towards an Underlying Network.

NOTE 2: The mapping by the Underlying Network of the M2M-Ext-ID to the M2M Device is Underlying Network specific.

NOTE 3: The Underlying Network provider and the M2M Service Provider collaborate for the assignment of an M2M-Ext-ID to each CSE identified by CSE-ID. At the same time, the Underlying Network provider maintains association of the M2M-Ext-ID with the Underlying Network specific Identifier allocated to the M2M device that hosts such CSE.

For pre-provisioned M2M-Ext-IDs, the M2M-Ext-ID along with the associated CSE-ID shall be made available at the Infrastructure Node. The CSE at M2M device does not need to have knowledge of the M2M-Ext-ID assigned to it.

For dynamic M2M-Ext-IDs, the M2M-Ext-ID specific to the Underlying Network shall be made available at the M2M device in the Field Domain. Such M2M-Ext-ID shall be conveyed to the IN-CSE during CSE Registration.

7.1.9 Underlying Network Identifier (UNetwork-ID)

The UNetwork-ID is used for identifying an Underlying Network. UNetwork-ID is a static value and unique within a M2M Service Provider domain.

One or more Underlying Networks may be available at an M2M Node offering different sets of capabilities, availability schedules etc. Based on the "policy" information at the Node and the capabilities offered by the available Underlying Networks, appropriate Underlying Network can be chosen by using UNetwork-ID. For example, based on "policy", scheduling of traffic triggered by a certain event category in certain time periods may be allowed over Underlying Network "WLAN" but may not be allowed over Underlying Network "2G Cellular".

7.1.10 Trigger Recipient Identifier (Trigger-Recipient-ID)

The Trigger-Recipient-ID is used when device triggering services are requested from the Underlying Network, to identify an instance of an ASN/MN-CSE on an execution environment, to which the trigger is routed. For example, when 3GPP device triggering is used, the Trigger-Recipient-ID maps to the Application-Port-Identifier (TS 123 682 [i.17]).

NOTE 1: For pre-provisioned M2M-Ext-IDs, Trigger-Recipient-ID is provisioned at the Infrastructure Node along with the M2M-Ext-ID and the associated CSE-ID.

NOTE 2: For dynamic M2M-Ext-IDs, Trigger-Recipient-ID specific to the Underlying Network is provisioned at each M2M device in the Field Domain. Such Trigger-Recipient-ID is conveyed to the IN-CSE during CSE Registration.

7.1.11 M2M Service Identifier (M2M-Serv-ID)

The M2M-Serv-ID is an identifier of a M2M Service offered by an M2M SP. It is an essential part of the M2M Service Subscription which stores a set of M2M-Serv-IDs pertaining to the set of subscribed services. Beyond the set of services depicted in this specification it shall be possible for an M2M Service Provider to offer other services. Those will be identified by means of M2M SP specific M2M-Serv-IDs.

7.1.12 M2M-SP-ID, CSE-ID and AE-ID and resource Identifier formats

As a general rule, the identifiers of AEs, CSEs and resources are globally unique. In order to optimize their use, the identifiers shall be shortened when their scope can be derived from their context of use by the CSEs and the AEs. Such shortened identifiers are defined as 'relative' identifiers.

TheM2M system shall use the identifiers M2M-SP-ID, CSE-ID and AE-ID and resource identifiers according to the formats and the rules specified in the following table (table 7.1.12-1).

Table 7.1.12-1: Identifiers formats and use

	Format Name	Format	Rule of use
AE-ID EQUIVALENT formats	CSE-Relative-AE-ID (with respect to the context of the CSE hosting the AE)	/<unique identifier inside the CSE>	On the Mca reference point: to refer to AEs that are hosted by the directly attached (Registrar) CSE. The hosting CSE is responsible for guaranteeing that the CSE-Relative-AE-ID is unique in the context of the hosting CSE.
	SP-relative -AE-ID (with respect to the context of the SP hosting the AE)	CSE-ID/<unique identifier inside the CSE>	On the Mca and Mcc reference points: to refer to AEs that are hosted by the M2M Service Provider domain to which AE is attached.
	Global-CSE-ID	M2M-SP-ID/CSE-ID/<unique identifier inside the CSE>	On the Mca and Mcc reference points: to refer to AEs that are hosted by a different M2M Service Provider domain with respect to the one to which the AE is attached and on the Mcc' reference point for all the AEs
CSE-ID EQUIVALENT formats	SP-Relative-CSE-ID (with respect to the context of the SP hosting the CSE)	/CSEBase	On the Mca and Mcc reference points: to refer to CSEs that are hosted by the same M2M Service Provider domain
	Global-CSE-ID	M2M-SP-ID/CSEBase	On Mca and Mcc reference points: to refer to CSEs that are hosted by different M2M Service Provider domains and on the Mcc' reference point for all the CSEs
M2M-SP-ID format	(always globally unique)	/FQDN	When The M2M-SP-ID is used, the FQDN format applies
Resource identifier (URI) equivalent formats	SP-Relative-resource-URI (with respect to the context of the M2M Service Provider hosting the CSE)	ResourceURI	On the Mca and Mcc reference points: to refer to resources that are hosted by the same M2M Service Provider domain
	Global-URI	M2M-SP-ID/ResourceURI	On Mca and Mcc reference points: to refer to resources that are hosted by a different M2M Service Provider domain and on the Mcc' reference point for all the resources

As a consequence, the hosting CSE shall convert the AEs global and relative identifier according to table 7.1.12-1 when a request is transmitted across the Mcc and Mca reference points.

As a consequence, the IN-CSE shall convert the AEs, CSEs and resource global and relative identifiers according to table 7.1.12-1when a request is transmitted across the Mcc’ reference point.

7.1.13 Service Role identifier (SRole-ID)

In each M2M Service, one or multiple M2M Service Role(s) shall be defined by the M2M Service Provider. An M2M Service Role is defined as a set of privileges pertaining to resource types which are associated with M2M Service. See Annex G for examples of M2M Service Provider defined Service Roles.

7.2 M2M Identifiers lifecycle and characteristics

Table 7.2-1: M2M Identifiers lifecycle and characteristics

Identifier	Assigned by	Assigned to	Assigned during	Lifetime	Uniqueness	Used during	Remarks
M2M Service Provider Identifier	Out of scope	AE, CSE	Out of scope	Out of scope	Global	Provisioning	
Application Entity Identifier	AE or Registrar CSE	AE	AE start-up	Application Entity Registration	Global	- Application Entity Registration - Security Context Establishment - All other operations initiated by the AE	Security requirements apply for Security Context Establishment
Application Identifier	Out of scope	Out of scope	Pre-provisioned	Out of scope	Specific to M2M service deployment	- Application Entity registration	
CSE Identifier	M2M SP	CSE	Security Provisioning	Life of the CSE	Global	- Information flows (clause 10) - Security Context Establishment	Security requirements apply for Security Context Establishment
M2M Node Identifier	Out of Scope	M2M Node hosting CSE	Pre-provisioned	Life of the M2M Node	Global	- Device Management	Needs to be Read Only
M2M Subscription Identifier	M2M SP, Out of Scope	Application Entities, and one or more CSEs belonging to the same M2M subscriber	At service signup	Life of the M2M Service Subscription with the M2M Service Provider	Global	- Charging and Information Recorded - Role based access control - Authentication	Multiple CSEs can be allocated the same M2M Subscription Identifier
M2M-Request-ID	**Mcc:** CSE **Mca:** Application Entity	A request initiated by an AE or CSE	**Mcc:** When a request is initiated by a CSE, or handling of a request received by a CSE. **Mca:** When a request is initiated by an AE	Equal to the lifetime of the Request and its corresponding Response	**Mcc:** Global **Mca:** Local or global	Requests and corresponding responses	
External Identifier	Jointly between the Underlying Network provider and M2M SP.	M2M Node belonging to a CSE that wants to utilize services of the Underlying Network.	Administrative Agreement.	Life of the CSE.	Local or global, decided by the specific Underlying Network provider	Requests initiated by a CSE over the Mcn reference point, where applicable.	**Pre-Provisioned Mode:** Made available at the Infrastructure Node. **Dynamic Mode:** Made available at M2M device. Conveyed to IN-CSE during CSE Registration.

This is a draft oneM2M document and should not be relied upon; the final version, if any, will be made available by oneM2M Partners Type 1.

Identifier	Assigned by	Assigned to	Assigned during	Lifetime	Uniqueness	Used during	Remarks
Underlying Network Identifier	M2M SP	Underlying Networks	Pre-provisioned	Life of the Underlying Network	Local to M2M SP domain	UL Network selection	
Trigger Recipient Identifier	Execution Environment	ASN/MN-CSE	ASN/MN-CSE start-up or wake-up	Life of the CSE	Execution Environment-wide	Device Triggering procedures, where applicable	**Pre-Provisioned Mode:** Made available at Infrastructure Node along with M2M-Ext-ID. **Dynamic Mode:** Made available at M2M device. Conveyed to IN-CSE during CSE Registration along with M2M-Ext-ID.
M2M Service Identifier	M2M Service Provider, Out of Scope	A service defined by the M2M Service Provider which consists of a set of functions defined by this document.	Out of Scope	Out of Scope	Local to the M2M Service Provider	For M2M Service Subscription	
SRole-ID	M2M Service Provider	M2M Service Provider sets the set of service roles that a subscriber can choose from. M2M service subscription lists one or multiple subscribed to service roles.	Out of Scope	Out of scope	M2M Service Provider	Access Control Policy, M2M service subscription management	

7.3 Addressing an Application Entity

7.3.1 Application Entity Addressing

In M2M communication, the goal of M2M addressing is to reach the CSE with which the target AE is registered, and ultimately the target AE on the M2M Node on which the target AE is resident. This principle applies to all Application Entities.

Reachability and routing from/to AEs on M2M Nodes is associated with the CSEs with which these AEs are registered, and the connectivity of such CSEs to the Underlying Networks. Reaching an AE shall be performed through reaching the CSE the AE is registered with. A CSE-PoA (CSE Point of Access) shall provide the set of information needed to reach a CSE from an Underlying Network perspective. Typically a CSE-PoA contains information that is resolved into a network address.

7.3.2 Application Entity Reachability

7.3.2.1 CSE Point of Access (CSE-PoA)

The CSE-PoA shall be used by the M2M System to communicate with a CSE on an M2M Node. Once communication with a CSE is achieved, an AE registered with that CSE can be reached as long as the AE can be uniquely identified.

The information included in the CSE-PoA as well as the refresh of the CSE-PoA, depends on the characteristics of the Underlying Network and an M2M Node's transport capabilities.

7.3.2.2 Locating Application Entities

Locating an AE is a two-step process as follows:

- **Step 1:** There is a need to locate the CSE where the AE is registered. Locating the CSE shall be accomplished as follows:
    - For AEs associated with ASNs/MNs/INs, the CSE-PoA of the ASN-CSE/MN-CSE/IN-CSE where the AE is registered shall be used.
    - For AEs associated with ADNs, the CSE-PoA of the MN-CSE/IN-CSE where the ADN is registered shall be used.
- **Step 2:** The CSE shall locate the appropriate AE using its Application Entity Identifier (AE-ID).

7.3.2.3 Usage of CSE-PoA by the M2M System

The CSE-PoA holds the information used by the M2M System to locate routing information for a CSE. This information shall be provided by the CSE at registration time. However, the routing information related to a CSE (and ultimately to the target AE) in an M2M System depends on the characteristic of the Underlying Network. This impacts the criteria for updating the CSE-PoA by the registered CSE, in addition to the regular CSE registration updates. The information to be conveyed as CSE-PoA needs to support Underlying Network specifics.

In general the addressing and routing information related to a CSE can be achieved when a static public IP address is assigned to and M2M Node and direct DNS address translation or dynamic DNS address translation is used.

In those circumstances, the CSE-PoA for a registered CSE shall have a URI conforming to RFC 3986 [i.12] as follows:

- URI = scheme:/fullyqualifieddomainname/path/; or
- URI = scheme://ip-address/path/.

The following clauses specify the information to be conveyed in the CSE-PoA by a registered CSE for various types of Underlying Networks, as well as the criteria for updating the CSE-PoA for the registered CSEs, in addition to the normal CSE registration refresh.

7.3.2.3.1 CSE-PoA related to CSEs associated with a Fixed Network

In this case the CSE-PoA for a registered CSE shall have a URI as described above. If the IP address is private, then the address is usually built based on the address of the related PPP protocol which is a public IP address. This in turn is mapped to the corresponding private address.

7.3.2.3.2 CSE-PoA related to CSEs associated with Mobile Networks

If the IP address for the registered CSE cannot be reliably used, and cannot be included in the CSE-PoA, then the CSE-PoA for the registered CSE shall include appropriate information as required by the respective Underlying Networks and supported by oneM2M.

Each Underlying Network shall need to specify the means for allowing an M2M SP to fetch the IP address associated with a CSE attaching to that Underlying Network and consequently the information to be included in the CSE-PoA for the registered CSE.

In the event that the M2M SP has connections to multiple Underlying Networks, there is a need to establish a binding between the registered CSE and the associated Underlying Network. That binding may be established through CSEs explicitly identifying the Underlying Network at registration/update time. Otherwise the M2M SP may derive the identity of the Underlying Network, e.g. by using the link, over which the registration arrived, store it and bind it to the registration information.

In the scenarios an M2M Node in mobile networks is not reachable by the previously known IP address and it supports SMS, the originating CSE can make use of SMS for device triggering mechanism to wake up the M2M Node to renew the IP addresses or perform specific functionalities.

To support this option, the CSE-PoA shall, on Mcn interface to the Underlying Networks supporting such an SMS for device triggering mechanism, include identification information of the CSE (such as the external identifier as defined by TS 123 682 [i.17] in the case of Tsp-based triggering, or MSISDN or any identifier used by triggering network APIs), and send the request to the Underlying Network via the mechanisms supported, such as Tsp, Tsms, Network APIs.

Annex B shows the 3GPP defined interfaces for machine type communication interfaces and example device triggering flows.

7.3.2.3.3 CSE-PoA to CSEs associated with multiple Underlying Networks

When an M2M Node attaches to a fixed network, the CSE-PoA for a registered CSE shall conform to the procedures associated with the fixed network.

When an M2M Node attaches to a mobile network, the CSE-PoA for a registered CSE shall conform to the procedures associated that mobile network.

If an M2M Node is already attached to an Underlying Network and attaches to another Underlying Network, the CSE may update its PoA information at the remote CSE.

7.3.3 Notification Re-targeting

7.3.3.1 Application Entity Point of Access (AE-PoA)

A Notify request to an AE is sent by targeting <AE> resource on a hosting CSE. If the hosting CSE verifies access control privilege of the Originator, the hosting CSE shall re-target the request to the address specified as AE-PoA (i.e. pointOfAccess attribute of <AE> resource). The AE-PoA may be initially configured in the <AE> resource when the AE registers to the Registrar CSE. If the <AE> resource does not contain an AE-PoA, an active communication link, if available, can be used for re-targeting. If neither of them is available, the request cannot be re-targeted to the AE.

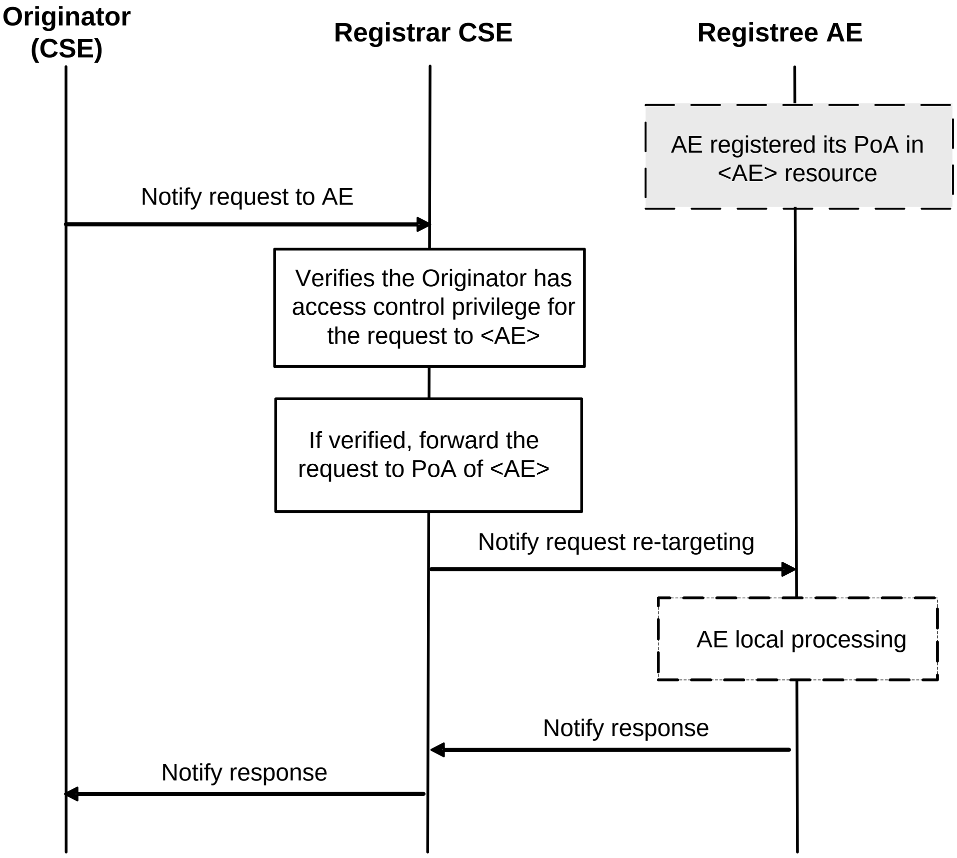

Originator (CSE)
Registrar CSE
Registree AE
AE registered its PoA in <AE> resource
Notify request to AE
Verifies the Originator has access control privilege for the request to <AE>
If verified, forward the request to PoA of <AE>
Notify request re-targeting
AE local processing
Notify response
Notify response

Figure 7.3.3-1: Re-targeting a notification request to an AE

8 Description and Flows of Reference Points

8.1 General Communication Flow Scheme on Mca and Mcc Reference Points

Procedures involving CSEs and AEs are driven by the exchange of messages across reference points according to the message flows described in this clause.

Depending on the message operation, procedures may manipulate information in a standardized resource structure as described in clause 9. Access and manipulation of the resources is subject to their associated privileges.

8.1.1 Description

Figure 8.1.1-1 shows the general flow that governs the information exchange within a procedure, which is based on the use of Request and Response messages. The message applies to communications such as:

- between an AE and a CSE (Mca reference point); and
- among CSEs (Mcc reference point).

Such communications can be initiated either by the AEs or by the CSEs depending upon the operation in the Request message.

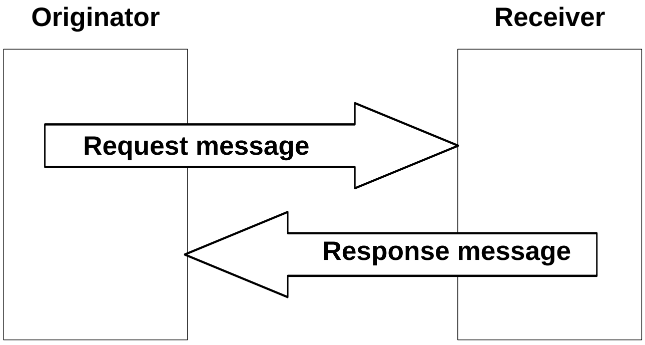


Figure 8.1.1-1: General Flow

8.1.2 Request

The Request from an Originator to a Receiver includes the following parameters:

- ***to*****:** URI of the target resource for the operation. The ***to*** parameter shall conform to clause 9.3.1

NOTE 1: **to** parameter can be known either by pre-provisioning (clause 11.2) or by discovery (clause 10.2.6 for discovery). Discovery of CSEBase is not supported in this release of the document. It is assumed knowledge of CSEBase is by pre-provisioning only.

NOTE 2: The term target resource refers to the resource which is addressed for the specific operation. For example the ***to*** parameter of a Create operation for a resource "example" would be "/m2m.provider.com/exampleBase". The ***to*** parameter for the Retrieve operation of the same resource "example" is "/m2m.provider.com/exampleBase/example".

- ***fr*****:** Identifier representing the Originator.

NOTE 3: The ***fr*** parameter shall be used by the Receiver to check the Originator identity for access privilege verification.

- ***cn*****:** resource content to be transferred.

- ***role:*** optional, required when role based access control is applied <associated text and procedure TBD>

[8.1.2.0a] Editor's Note: Need to figure out the impact of Role in different procedures.

- ***op*****:** operation to be executed: Create (C), Retrieve (R), Update (U), Delete (D), Notify (N)

The ***op*** parameter shall indicate the operation to be executed at the Receiver:

- **Create (C):** ***to*** is the URI of the target resource where the new resource (parent resource)
- **Retrieve (R):** an existing ***to*** addressable resource is read and provided back to the Originator.
- **Update (U):** the content of an existing ***to*** addressable resource is replaced with the new content as in ***cn*** parameter. If some attributes in the ***cn*** parameter do not exist at the target resource, such attributes are created with the assigned values. If some attributes in the ***cn*** parameter are set to NULL, such attributes are deleted from the addressed resource.
- **Delete (D):** an existing ***to*** addressable resource and all its sub-resources are deleted from the Resource storage.
- **Notify (N):** information to be sent to the Receiver, processing on the Receiver is not indicated by the Originator.

The ***ty*** parameter shall be present in Request for the following operations:

- **Create:** ***ty*** is the type of the resource to be created.
- ***cn*****:** resource content to be transferred.

The ***cn*** parameter shall be present in Request for the following operations:

- **Create:** ***cn*** is the content of the new resource with the resource type ***ty.***
- **Update:** ***cn*** is the content to be replaced in an existing resource. For the Update operation that is used for Execute operation, ***cn*** does not exist. For attributes to be created at the resource, ***cn*** includes names of such attributes with their associated values. For attributes to be deleted at the resource, ***cn*** includes the names of such attributes with their value set to NULL.
- **Notify:** ***cn*** is the notification information.

Note that the ***cn*** parameter can also be empty.

Other allowed parameters shall be as follows:

- ***nm*****:** optional name of the resource to be created.

  Example usage of the name includes a name that the Originator of the Create resource wishes to be used as the identifier of the newly created resource. For creating a container with the name "myContainer" the request will provide the parameter ***nm*** with the value "myContainer" and the created resource would be: /<CSEBase>/myContainer.

- ***ot*****:** optional originating timestamp of when the message was built.

  Example usage of the originating timestamp includes: to measure and enable operation (e.g. message logging, correlation, message prioritization/scheduling, accept performance requests, charging, etc.) and to measure performance (distribution and processing latency, closed loop latency, SLAs, analytics, etc.)

- ***rqet*****:** optional request message expiration timestamp.

  Example usage of the request expiration timestamp includes to indicate when request messages (including delay-tolerant) should expire due to their staleness being no longer of value, and to inform message scheduling/prioritization. When a request has set request expiration timestamp to a specific time and the Request demands an operation on a hosting CSE that is not the CSE currently processing the request, then the current CSE shall keep on trying to deliver the Request to the hosting CSE until the request expiration timestamp time, in line with provisioned policies.

- ***rset*****:** optional result message expiration timestamp.

Example usage of the result expiration timestamp includes to indicate when result messages (including delay-tolerant) should expire due to expected staleness of the result, being no longer of value, and to inform message scheduling/prioritization. It can be used to set the maximum allowed total request/result message sequence round trip deadline.

- ***rt*:** optional response message type: Indicates what the response to the issued request shall contain and when the response is sent to **t**he Originator:

  - nonBlockingRequestSynch: In case the request is accepted by the Local CSE, the Local CSE responds after acceptance with an Acknowledgement confirming that the Local CSE will further process the request. The Local CSE includes in the response to an accepted request a reference that can be used to access the status of the request and the result of the requested operation at a later time. Processing of Non-Blocking Requests is defined in clause 8.2.2 and in particular for the synchronous case in clause 8.2.2.2.

  - nonBlockingRequestAsynch: In case the request is accepted by the Local CSE, the Local CSE responds after acceptance with an Acknowledgement confirming that the Local CSE will further process the request. The result of the requested operation needs to be sent as a notification to the Originator. Processing of Non-Blocking Requests is defined in clause 8.2.2 and in particular for the asynchronous case in clause 8.2.2.3.

  - blockingRequest: In case the request is accepted by the Local CSE, the Local CSE responds with the result of the requested operation after completion of the requested operation. Processing of Blocking Requests is defined in clause 8.2.1.

  Example usage of the response type set to nonBlockingRequestSynch: An Originator that is optimized to minimize communication time and energy consumption wants to express a Request to the local CSE and get an acknowledgement on whether the Request got accepted. After that the Originator may switch into a less power consuming mode and retrieve a Result of the requested Operation at a later time.

  Further example usage of response type set to nonBlockingRequestSynch includes when the result content is extremely large, or when the result consists of multiple content parts from a target group which are to be aggregated asynchronously over time.

- ***rc*:** optional result content: Indicates what are the expected components of the result of the requested operation. The Originator of a request may not need to get back a result of an operation at all. This shall be indicated in the ***rc*** parameter. Which exact settings of ***rc*** are possible depends on the requested operation specified in ***op***. Possible values of ***rc*** are:

  - **attributes:** Representation of the requested resource shall be returned as content, without the URI(s) of the child resource(s). This is the default value. For example, if the request is to retrieve a <container> resource, the URI(s) of the <contentInstance> child-resource(s) is not provided. This setting is not valid for a Notify operation.

  - **attributes+child-resources:** Representation of the requested resource, along with the URI(s) of the child resource(s), possibly limited by a maximum number of retrieved links, shall be returned as content. For example, if the request is to retrieve a <container> resource, the <container> resource and the URI(s) of the <contentInstance> child-resource(s) are provided. This setting is not valid for a Notify operation

  - **child-resource:** URI(s) of the child resources, possibly limited by a maximum number of retrieved URI(s), without any representation of the actual requested resource shall be returned as content. For example, if the request is to retrieve a <container> resource, only the URI(s) of the <contentInstance> child-resource(s) is provided. This setting is not valid for a Notify operation

  - **nothing:** Nothing shall be returned as operational result content. This setting is not valid for a retrieve operation. This setting is the default when a Notification was requested by the **op** parameter.

- **original-resource:** Representation of the original resource pointed by the *link* attribute in the announced resource shall be returned as content, without the URI(s) of the child resource(s). This setting is only valid when the to parameter in the RETRIEVE Request targets the announced resource.

[8.1.2.f0] Editor's Note: Example usage of all **rc** enumerated values needed. What is the use case for "nothing"?

[8.1.2.f1] Editor's Note: Do we need to support use case for "no response expected" in this release.

- ***rp*****:** optional response persistence: indicates the duration for which the address containing the responses is to persist.

  Example usage of response persistence includes requesting sufficient persistence for analytics to process the response content aggregated asynchronously over time. If a result expiration timestamp is specified then the response persistence should last beyond the result expiration time.

- ***ri*****:** request Identifier.

  Example usage of request identifier includes enabling the correlation between a Request and one of the many received Responses.

- ***oet*****:** optional operation execution time: indicates the time when the specified operation ***op*** is to be executed by the target CSE. A target CSE shall execute the specified operation of a Request having its operational execution time indicator set, starting at the operational execution time. If the execution time has already passed or if the indicator is not set, then the specified operation shall be immediately executed, unless the request expiration time, if set, has been reached.

  Example usage of operational execution time includes asynchronous distribution of flows, which are to be executed synchronously at the operational execution time.

NOTE 4: Time-based flows could not supported depending upon time services available at CSEs.

- ***ec*****:** optional event category: Indicates the event category that should be used to handle this request. Event categories are impacting how Requests to access remotely hosted resources are processed in the CMDH CSF. Selection and scheduling of connections via CMDH are driven by policies that can differentiate event categories.

  Example usage of "event category" set to specific value X: When the request is demanding an operation to be executed on a hosting CSE that is different from the Registrar CSE, the request may be stored in the CSE that is currently processing the request on the way to the hosting CSE until it is allowed by provisioned policies for that event category X to use a communication link to reach the next CSE on a path to the hosting CSE or until the request expiration timestamp is expired.

  The following values for '***ec***' shall have a specified pre-defined meaning:

  - ***ec*** **= 'immediate':** Requests of this category shall be sent as soon as possible and shall not be subject to any further CMDH processing, i.e. the request will not be subject to storing in CMDH buffers when communication over an underlying network is possible. In particular, CMDH processing will respect values for ***rqet***, ***rset*** given in the original request and not fill in any default values if they are missing.

  - ***ec*** **= 'bestEffort':** Requests of this category can be stored in CDMH buffers at the discretion of the CSE that is processing the request for an arbitrary time and shall be forwarded via Mcc on a best effort basis. The CSE does not assume any responsibility to meet any time limits for delivering the information to the next CSE. Also the maximum amount of buffered requests for this category is at the discretion of the processing CSE.

  - ***ec*** **= 'latest':**

    - If this category is used in a request asking for a CRUD operation on a resource, the following shall apply:
      CRUD requests using this category shall undergo normal CMDH processing as outlined further below in this specification and in [i.2] with a maximum buffer size of one pending request for a specific pair of ***fr*** and ***to*** parameters that appear in the request. If a new request message is received by the CSE with a pair of parameters ***fr*** and ***to*** that has already been buffered for a pending request, the newer request will replace the buffered older request.

    - If this category is used in a notification request triggered by a subscription, the following shall apply:
      Notification requests triggered by a subscription using this category shall undergo normal CMDH processing as outlined further below in this specification and in [i.2] with a maximum buffer size of one pending notification request per subscription reference that appears in a notification request. If a new notification request is received by the CSE with a

subscription reference that has already been buffered for a pending notification request, the newer request will replace the buffered older request.

If no further CMDH policies are provisioned for this event category, the forwarding process shall follow the ‘bestEffort’ rules defined above.

The M2M Service Provider shall be able to provision CMDH policies describing details for the usage of the specific Underlying Network(s) and the applicable rules as defined in the [cmdhPolicy] resource type for other ***ec*** values not listed above.

- ***da*****:** optional delivery aggregation on/off: Use CRUD operations of <delivery> resources to express forwarding of one or more original requests to the same target CSE(s).

NOTE 5: Since ***da*** is optional, there could be a default value to be used when not present in the Request. This parameter could not be exposed to AEs via Mca.

Example usage of delivery aggregation set on: The CSE processing a request shall use aggregation of requests to the same target CSE by requesting CREATE of a <delivery> resource on the next CSE on the path to the target CSE.

- ***gid*****:** optional group request identifier: Identifier optionally added to the group request that is to be fanned out to each member of the group in order to detect loops and avoid duplicated handling of operation in case of loops of group and common members between groups that have parent-child relationship.

- ***fc*****:** optional filter criteria: conditions for filtered retrieve operation are described in table 8.1.2-1. This is used for resource discovery (clause 10.2.6) and general retrieve, update, delete requests (clause 10.1.2, 10.1.3 and 10.1.4).

Example usage of retrieve requests with filter criteria using modifiedSince condition tag: if a target resource is modified since 12:00 then the hosting CSE will send a resource representation.

- ***Disrestype:*** Optional Discovery result type. This parameter applies to discovery related requests (see clause 10.2.6) to indicate the preference of the Originator for the format of returned information in the result of the operation. This parameter shall take on one of the following values reflecting the options in clause 9.3.1:

  - Hierarchical URI option.
  - Non-hierarchical URI option.
  - CSE-ID, and resource identifier option.

Example usage of the discovery result type set to Non-hierarchical URI option. The request originator is only willing to receive non-hierarchal URIs for discovered resources.

The absence of the parameter implies that the result shall be in Hierarchical URI format.

[8.1.2.g] Editors Note: This may be impacted by the management of access control policy which will be enhanced to take into account Hierarchical and non-Hierarchical URI.

Rapportuer Note: Discussions during ARC#10bis. Contribution expected for the resolution of the above stated Editor's Note.

Table 8.1.2-1: *filterCriteria* conditions

Condition tag	Multiplicity	Matching condition
createdBefore	0..1	The *creationTime* attribute of the resource is chronologically before the specified value.
createdAfter	0..1	The *creationTime* attribute of the resource is chronologically after the specified value.
modifiedSince	0..1	The *lastModifiedTime* attribute of the resource is chronologically after the specified value.
unmodifiedSince	0..1	The *lastModifiedTime* attribute of the resource is chronologically before the specified value.
stateTagSmaller	0..1	The *stateTag* attribute of the resource is smaller than the specified value.
stateTagBigger	0..1	The *stateTag* attribute of the resource is bigger than the specified value.
expireBefore	0..1	The *expirationTime* attribute of the resource is chronologically before the specified value.
expireAfter	0..1	The *expirationTime* attribute of the resource is chronologically after the specified value.
labels	0..n	The *labels* attributes of the resource matches the specified value.
resourceType	0..n	The *resourceType* attribute of the resource is the same as the specified value. It also allows discriminating between normal and announced resources.
sizeAbove	0..1	The *contentSize* attribute of the <contentInstance> resource is equal to or greater than the specified value.
sizeBelow	0..1	The *contentSize* attribute of the <*contentInstance*> resource is smaller than the specified value.
contentType	0..n	The *typeOfContent* attribute of the <*contentInstance*> resource matches the specified value.
limit	0..1	Limitation the number of matching resources to the specified value.
attribute	0..n	This is an attribute of resource types (clause 9.6). Therefore, a real tag name is variable and depends on its usage. E.g., *creator* of container resource type can be used as a filter criteria tag as “creator=Sam”.
filterUsage	0..1	Indicates how the filter criteria is used. E.g., if this parameter is not provided, the Retrieve operation is for generic retrieve operation. If filterUsage is provided, the Retrieve operation is for resource <discovery> (clause 10.2.6).

Example usage of filter criteria conditions in a HTTP query: an HTTP GET operation can be requested applying also a filter in the query part of the request itself:

GET /root?label=one&label=two&createdBefore=2014-01-01T00:00:00&limit=128&filterUsage=discovery

The example discovers a maximum of 128 resources matching the following logical condition: createdBefore < 2014-01-01T00:00:00 AND (label = one OR label = two).

Once the Request is delivered, the Receiver shall analyze the Request to determine the target resource.

If the target resource is addressing another M2M node, the Receiver shall route the request appropriately.

If the target resource is addressing the Receiver, it shall:

- Check the existence of ***to*** addressed resource.
- Identify the resource type by ***ty***.
- Check the privileges for ***fr*** Originator to perform the requested operation.
- Perform the requested operation (using ***cn*** content when provided) according to the provided request parameters as described above.

- Depending on the request result content, respond to the Originator with indication of successful or unsuccessful operation results. In some specific cases (e.g. limitation in the binding protocol or based on application indications), the Response could be avoided.

The message flow procedure started with an Originator Request message shall be considered closed when either:

- A Request message is expired according to the ***rqet*** (request expiration timestamp).

- A Response message is delivered to the Originator.

8.1.2.1 Summary of Request Message Parameters

Table 8.1.2.1-1summarises the parameters specified in clause 8.1.2 for the Request message, showing any differences as applied to C, R, U, D or N operations. "M" indicates mandatory, "O" indicates optional, "N/A" indicates "not applicable".

Table 8.1.2.1-1: Summary of Request Message Parameters

Request message parameter\Operation	Create	Retrieve	Update	Delete	Notify
Operation (*op*) - operation to be executed	M	M	M	M	M
To (*to*) - the address of the target resource on the target CSE	M	M	M	M	M
From (*fr*) - the identifier of the message Originator	M	M	M	M	M
Request Identifier (*ri*) - uniquely identifies a Request message	M	M	M	M	M
Resource Type (*ty*) - of resource to be created	M	N/A	N/A	N/A	N/A
Name (*nm*) - of resource to be created	O	N/A	N/A	N/A	N/A
Content (*cn*) - to be transferred	M	O	Conditional M (this parameter is mandatory for all UPDATEs other than those used for execute operation for device management. See clause 10.2.7.6.).	N/A	M
Originating Timestamp (*ot*) - when the message was built	O	O	O	O	O
Request Expiration Timestamp (*rqet*) - when the request message expires	O	O	O	O	O
Result Expiration Timestamp (rset) - when the result message expires	O	O	O	O	O
Operational Execution Time (*oet*) - the time when the specified operation is to be executed by the target CSE.	O	O	O	O	O
Response Type (*rt*) - Registrar CSE response shall either indicate that the Request was accepted, or include the operation result	O	O	O	O	O
Result Persistence (*rp*) - the duration for which the reference containing the responses is to persist	O	O	O	O	N/A
Result Content (*rc*) - the expected components of the result	O	O	O	O	N/A
Event Category (*ec*) - indicates how and when the system should deliver the message	O	O	O	O	O
Delivery Aggregation (*da*) - aggregation of requests to the same target CSE is to be used	O	O	O	O	O
Group Request Identifier (gid) - Identifier added to the group request that is to be fanned out to each member of the group.	O	O	O	O	O
Filter Criteria (*fc*) - conditions for filtered retrieve operation	N/A	O	N/A	N/A	N/A
Discovery Result Type (Disrestype) - format of information returned for Discovery operation	N/A	O	N/A	N/A	N/A

8.1.3 Response

The Response received by the Originator of a Request accessing resources over the **Mca** and **Mcc** reference points shall contain mandatory and may contain optional parameters. Certain parameters may be mandatory or optional depending upon the Requested operation (CRUDN) or the mandatory response code. In this clause, the mandatory parameters are detailed first, following by those that are conditional, followed by those that are optional:

Mandatory Parameters:

- ***rs*:** response code: This parameter indicates whether the operation was successful, unsuccessful or is an acknowledgement:
  - A "successful" code indicates to the Originator that the Requested operation has been executed successfully by the hosting CSE.
  - An "unsuccessful" code indicates to the Originator that the Requested operation has not been executed successfully by the hosting CSE.
  - An "acknowledgement" indicates to the Originator that the Request has been received and accepted by the attached CSE, i.e. by the CSE that received the Request from the issuing Originator directly, but the Request operation has not been executed yet. The success or failure of the execution of the Requested operation is to be conveyed later.

  Details of successful, unsuccessful, and acknowledge codes are provided in the oneM2M Core Protocol Technical Specification (TS-0004 [i.2]).

- ***ri*:** Request Identifier. The ***ri*** in the Response shall match the ***ri*** in the corresponding Request.

Conditional Parameters:

- ***cn*:** resource content:
  - If ***rs*** is "successful" then:

    The ***cn*** parameter may be present in a Response in the following cases:

    - **Create:** ***cn*** is the address and/or the content of the created resource.
    - **Update:** ***cn*** is the content replaced in an existing resource. If attributes are created at an existing resource, ***cn*** includes the names of attributes created and their associated values. If attributes are deleted at an existing resource, ***cn*** includes the names of the attributes deleted.
    - **Delete:** ***cn*** is the content actually deleted.

    The ***cn*** parameter shall be present in a Response in the following cases:

    - **Retrieve:** ***cn*** is the retrieved resource content or aggregated contents of discovered resources.

      If present in the Request, result contents ***rc***, indicates which components of the result of the requested operation are to be included in the Response.

  - If ***rs*** is "unsuccessful" then ***cn*** may be present in a Response to provide more error information.
  - If ***rs*** is "acknowledgment" then ***cn*** is not present.

Optional parameters:

- ***to*:** ID of the Originator.
- ***fr*:** ID of the Receiver.
- ***ot*:** originating timestamp of when the message was built.
- ***rset*:** result expiration timestamp. The Receiver shall echo the result expiration timestamp if set in the Request message, or may set the result expiration timestamp itself.

Example usage of the Receiver setting the result expiration timestamp is when the value of the delivery time is dependent upon some changing Receiver context e.g. Result message deadline for aircraft position based upon velocity.

- ***cs*:** status codes (e.g. authorization timeout, etc.).

8.1.4 Summary of Response Message Parameters

Table 8.1.4-1 summarises the parameters specified in clauses 1.3 for the Response messages, showing any differences as applied to successful C, R, U, D or N operations, and unsuccessful operations. "M" indicates mandatory, "O" indicates optional, "N/A" indicates "not applicable".

Table 8.1.4-1: Summary of Response Message Parameters

Response message parameter \ success or not	rs = Ack	rs = successful: Op = Create	rs= successful: Op= Retrieve	rs= successful. Op= Update	rs= successful. Op = Delete	rs= successful Op= Notify	rs = unsuccessful Op = C,R,U,D or N
Response Code (*rs*) - successful, unsuccessful, ack	M	M	M	M	M	M	M
Request Identifier (*ri*) - uniquely identifies a Request message	M	M	M	M	M	M	M
Content (*cn*) - to be transferred	N/A	O (The address and/or the content of the created resource)	M (the retrieved resource content or aggregated contents of discovered resources)	O (The content replaced in an existing resource. The content of the new attributes created. The name of the attributes deleted.)	O (The content actually deleted)	N/A	O (Additional error info)
To (*to*) -the identifier of the Originator	O	O	O	O	O	O	O
From (*fr*) - the identifier of the Receiver	O	O	O	O	O	O	O
Originating Timestamp (*ot*) - when the message was built	O	O	O	O	O	O	O
Result Expiration Timestamp (*rset*) - when the message expires	O	O	O	O	O	N/A	O
Status codes (*cs*) - (e.g. authorization timeout, etc.)	O	O	O	O	O	O	O
Response Address (*ra*) - address for the temporary storage of end node Responses.	O	O	O	O	O	N/A	N/A

8.2 Procedures for Accessing Resources

This clause describes the procedures for accessing the resources. The term "hop" in the descriptions here refers to the number of transit CSEs that forward a request from the Originator CSE to the hosting CSE.

All the descriptions and message flows in this clause are illustrative for the direction from a Registree acting as an Originator to a Registrar acting as a Receiver only. The flows from a Registrar CSE to a Registree CSE are symmetric with respect to the one described in this section. Both IN-CSE and MN-CSE have ability to route a received request or response messages to one of its Registrees. If the hosting CSE is not known by an MN-CSE that receives a request or response message, that MN-CSE shall forward the message to its own Registrar CSE by default.

8.2.1 Accessing Resources in CSEs - Blocking Requests

For the procedures described herein, the addressed resource can be stored in different CSEs. Table 8.2.1-1 describes the possible scenarios, where the addressed resource may be on the Registrar CSE or on a CSE located elsewhere in the oneM2M System.

In this clause - for simplicity - it is assumed that the Originator of a Request can always wait long enough to get a Response to the Request after the requested operation has finished. This implies potentially long or unknown blocking times (time for which a pending Request has not been responded to) for the Originator of a Request.

For scenarios that avoid such possibly long blocking times, clause 8.2.2 specifies mechanisms to handle synchronous and asynchronous resource access procedures via returning appropriate references.

Table 8.2.1-1: Accessing Resources in different CSEs, from Registree to Registrar CSE

Number of Transit CSEs	Description	Reference
No Hops	• The Originator of the Request accesses a resource. • The Originator of the Request can be an AE or a CSE. • Registrar CSE and hosting CSE are the same entity. • The hosting CSE checks the Access Control Privileges for accessing the resource. • Depending on the expected result content, the hosting CSE responds to the Originator of the Request, either with a success or failure Response	Figure 8.2.1-1
1 Hop	• The Originator of the Request accesses a resource. • The Originator of the Request may be an AE or a CSE. • Registrar CSE and hosting CSEs are different entities. • Registrar CSE forwards the Request to the hosting CSE if the Registrar CSE is registered with the hosting CSE, for accessing the resource • Hosting CSE checks the Access Control Privileges for accessing the resource and depending on the expected result content respond with a success or failure Response.	Figure 8.2.1-2
Multi Hops	• The Originator of the Request accesses a resource. • The Originator of the Request may be an AE or a CSE. • Registrar CSE, Transit CSE(s) and the hosting CSE are different entities. • Registrar CSE: – Forwards the Request to a Transit-1 CSE (e.g. MN-CSE) that the Registrar CSE is registered with, if configured through policies to do so or; – Forwards the request to an IN-CSE if the Registrar CSE is registered with IN-CSE and if configured through policies to do so. • Transit-N CSE: – Forwards the request to the hosting CSE if it is registered with the hosting CSE or – Forwards the Request to another Transit-(N+1) CSE (e.g. another MN-CSE) that the Transit-N CSE is registered with. or – Forwards the request to an IN-CSE if the Transit-N CSE is registered with the IN-CSE. • In case the Request reaches the IN-CSE, the IN-CSE: – Performs the processing defined under 'hosting CSE' below if the targeted resource is hosted on IN-CSE. – Forwards the request to another IN-CSE if the resource belongs to another M2M SP or – Forwards the request to the hosting CSE if the latter is known (e.g. announcements) by the IN-CSE. • Hosting CSE checks the Access Control Privileges for accessing the resource and depending on the expected result content respond with a success or failure Response.	Figure 8.2.1-3

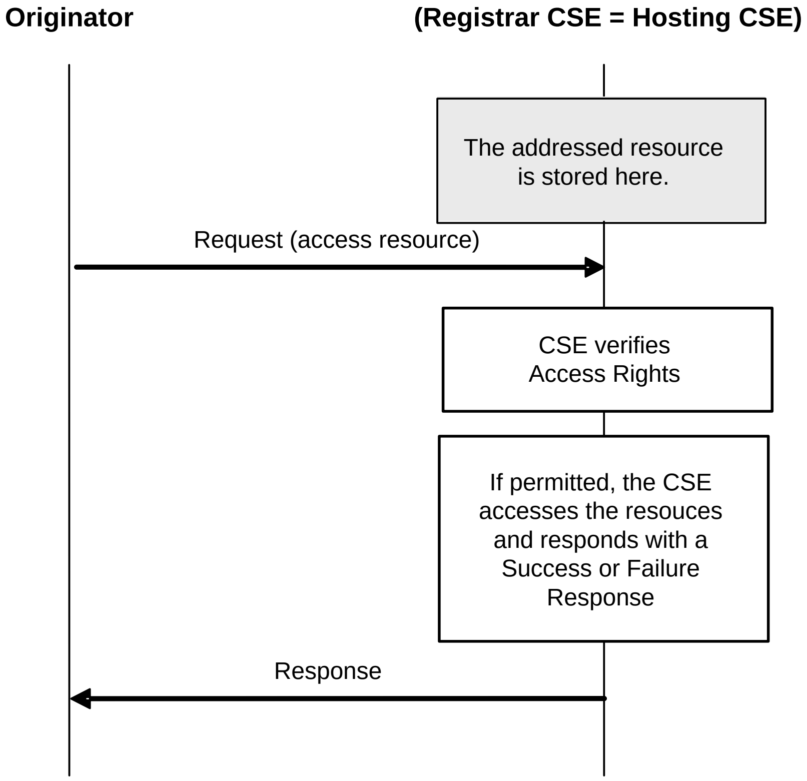

Originator
(Registrar CSE = Hosting CSE)
The addressed resource is stored here.
Request (access resource)
CSE verifies Access Rights
If permitted, the CSE accesses the resouces and responds with a Success or Failure Response
Response

Figure 8.2.1-1: Originator accesses a resource on the Registrar CSE (No Hops)

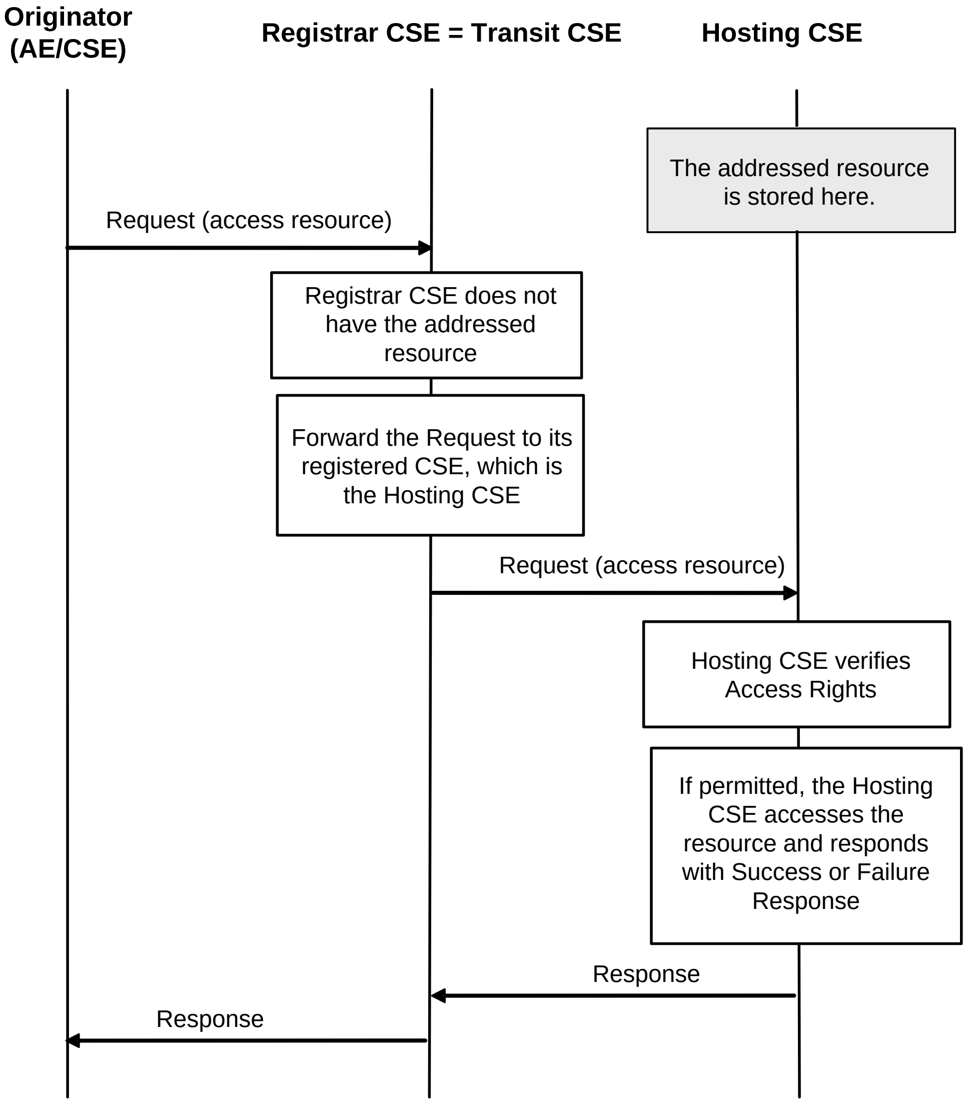

Originator (AE/CSE)
Registrar CSE = Transit CSE
Hosting CSE
The addressed resource is stored here.
Request (access resource)
Registrar CSE does not have the addressed resource
Forward the Request to its registered CSE, which is the Hosting CSE
Request (access resource)
Hosting CSE verifies Access Rights
If permitted, the Hosting CSE accesses the resource and responds with Success or Failure Response
Response
Response

Figure 8.2.1-2: AE/CSE accesses a resource at the Hosting CSE (One Hop)

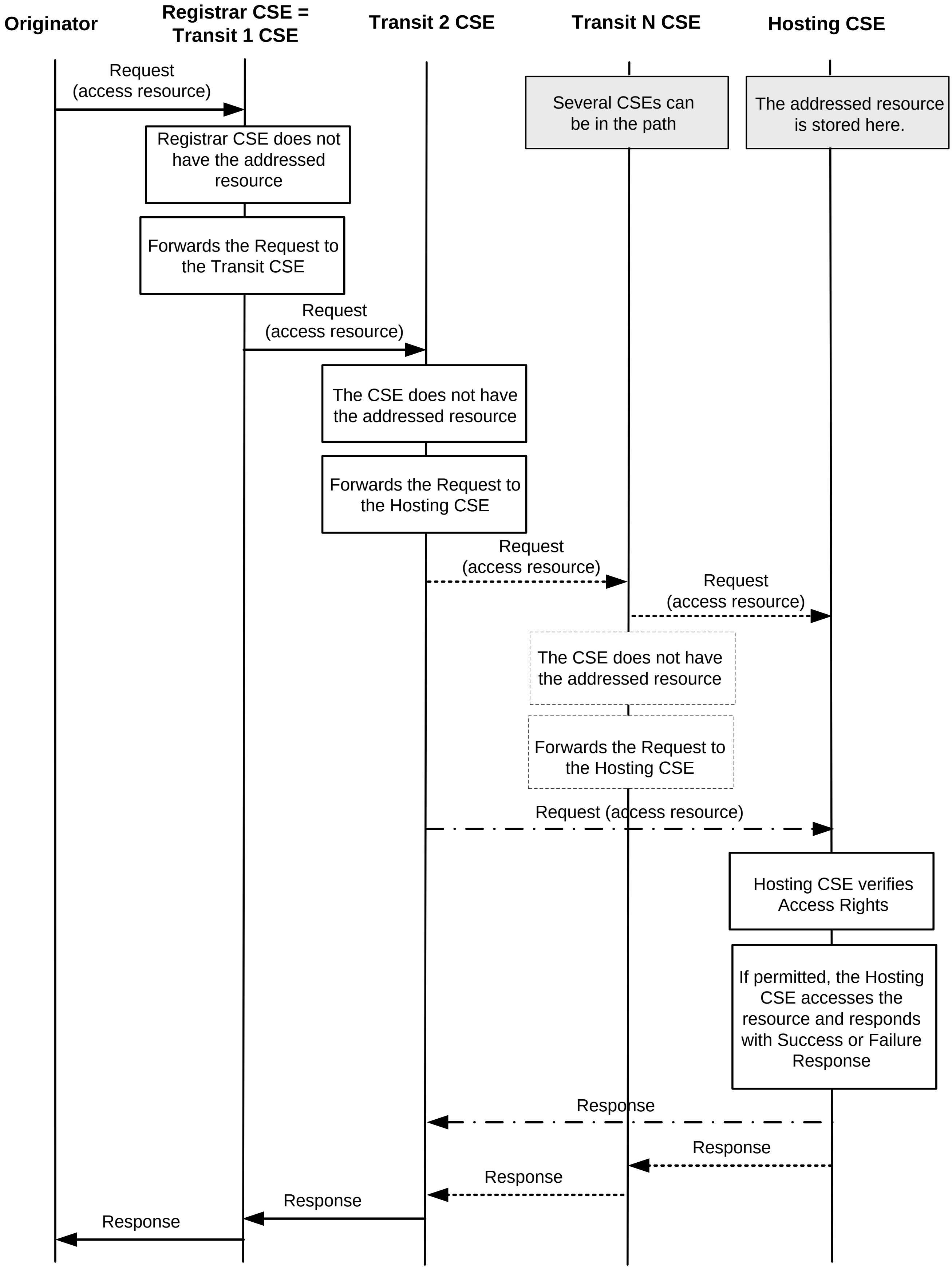

Originator
Registrar CSE = Transit 1 CSE
Transit 2 CSE
Transit N CSE
Hosting CSE
Several CSEs can be in the path
The addressed resource is stored here.
Request (access resource)
Registrar CSE does not have the addressed resource
Forwards the Request to the Transit CSE
Request (access resource)
The CSE does not have the addressed resource
Forwards the Request to the Hosting CSE
Request (access resource)
Request (access resource)
The CSE does not have the addressed resource
Forwards the Request to the Hosting CSE
Request (access resource)
Hosting CSE verifies Access Rights
If permitted, the Hosting CSE accesses the resource and responds with Success or Failure Response
Response
Response
Response
Response
Response

Figure 8.2.1-3: Originator accesses a resource at the Hosting CSE (Multi Hops)

8.2.1.1 M2M Requests Routing Policies

CSEs can use policies to govern routing of M2M requests to the next hop towards its target. Routing, through these policies, can be based, for example, on the target CSE, target M2M domain, specific types of resources if applicable, priority of a request, etc.

These policies are not defined in this release of this document. It is the responsibility of M2M SP and the CSE administrator to ensure the appropriateness of these policies for routing purposes.

8.2.2 Accessing Resources in CSEs - Non-Blocking Requests

8.2.2.1 Response with Reference to Result of Requested Operation

In case the Originator of a Request has asked for only a response with an Acknowledgement of the Request and a reference to the result of the requested operation - see ***rt*** parameter in clause 8.1.2 - it is necessary to provide a prompt response to the Originator with a reference to an internal resource on the Registrar CSE or another specified reference, so that the Originator can retrieve the outcome of the requested operation at a later time. The details of such an internal resource are defined in clause 9.6.12. The reference is provided in the response to the Request. The abbreviation "Req-Ref" is used for simplicity in the figures of the following clauses.

Two different cases to allow the Originator of a request to retrieve the result of a requested operation are defined in the following two clauses.

8.2.2.2 Synchronous Case

In the synchronous case, it is assumed that the Originator of a Request is not able to receive asynchronous messages, i.e. all exchange of information between Originator and Registrar CSE needs to be initiated by the Originator.

In that case the information flow depicted in figure 8.2.2.2-1 is applicable. For the flow depicted in figure 8.2.2.2-1 it is assumed that completion of the requested operation happens before the Originator is trying to retrieve the result of the requested operation with a second request referring to the "Req-Ref" provided in the Response to the original Request.

Another variation of the information flow for the synchronous case is depicted in figure 8.2.2.2-2. In this variation it is assumed that the requested operation completes after the second request but before the third request sent by the Originator.

Equivalent information flows are valid also for cases where the target resource of the requested operation is not hosted on the Registrar CSE. From an Originator's perspective there is no difference as the later retrieval of the result of a requested operation would always be an exchange of Request/Response messages between the Originator and the Registrar CSE using the reference to the original request.

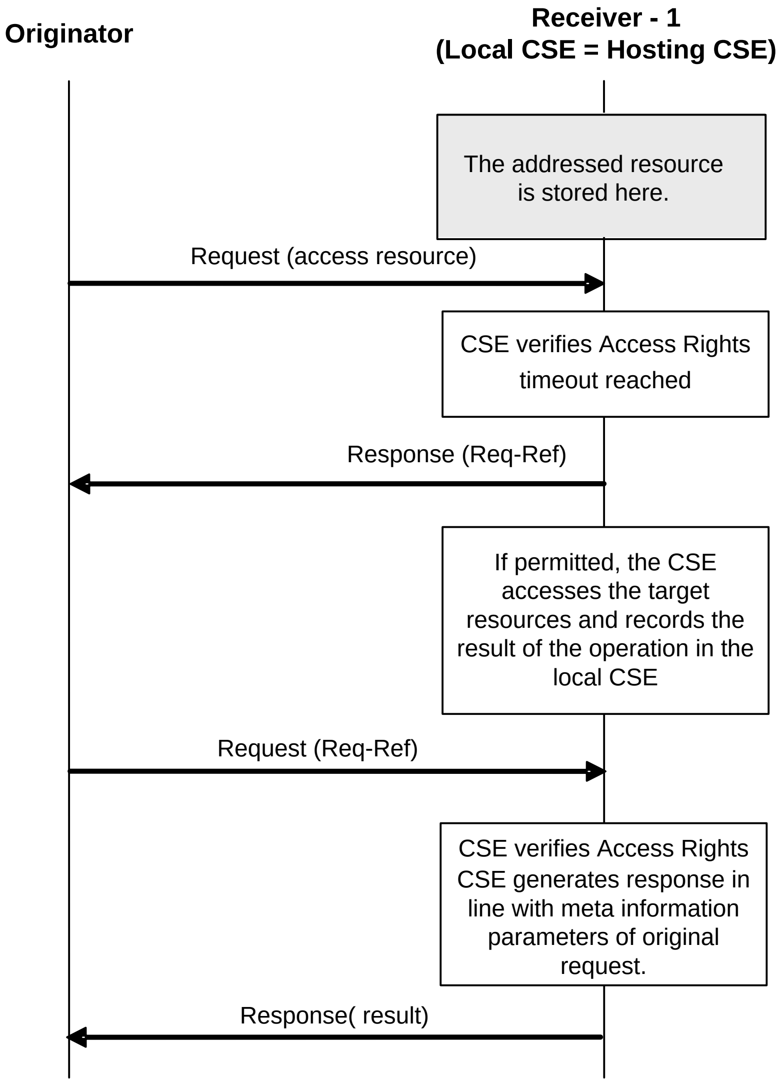

Originator
Receiver - 1
(Local CSE = Hosting CSE)
The addressed resource is stored here.
Request (access resource)
CSE verifies Access Rights
timeout reached
Response (Req-Ref)
If permitted, the CSE accesses the target resources and records the result of the operation in the local CSE
Request (Req-Ref)
CSE verifies Access Rights
CSE generates response in line with meta information parameters of original request.
Response(result)

Figure 8.2.2.2-1: Non-blocking access to resource in synchronous mode (Hosting CSE = Registrar CSE), requested operation completed before second request

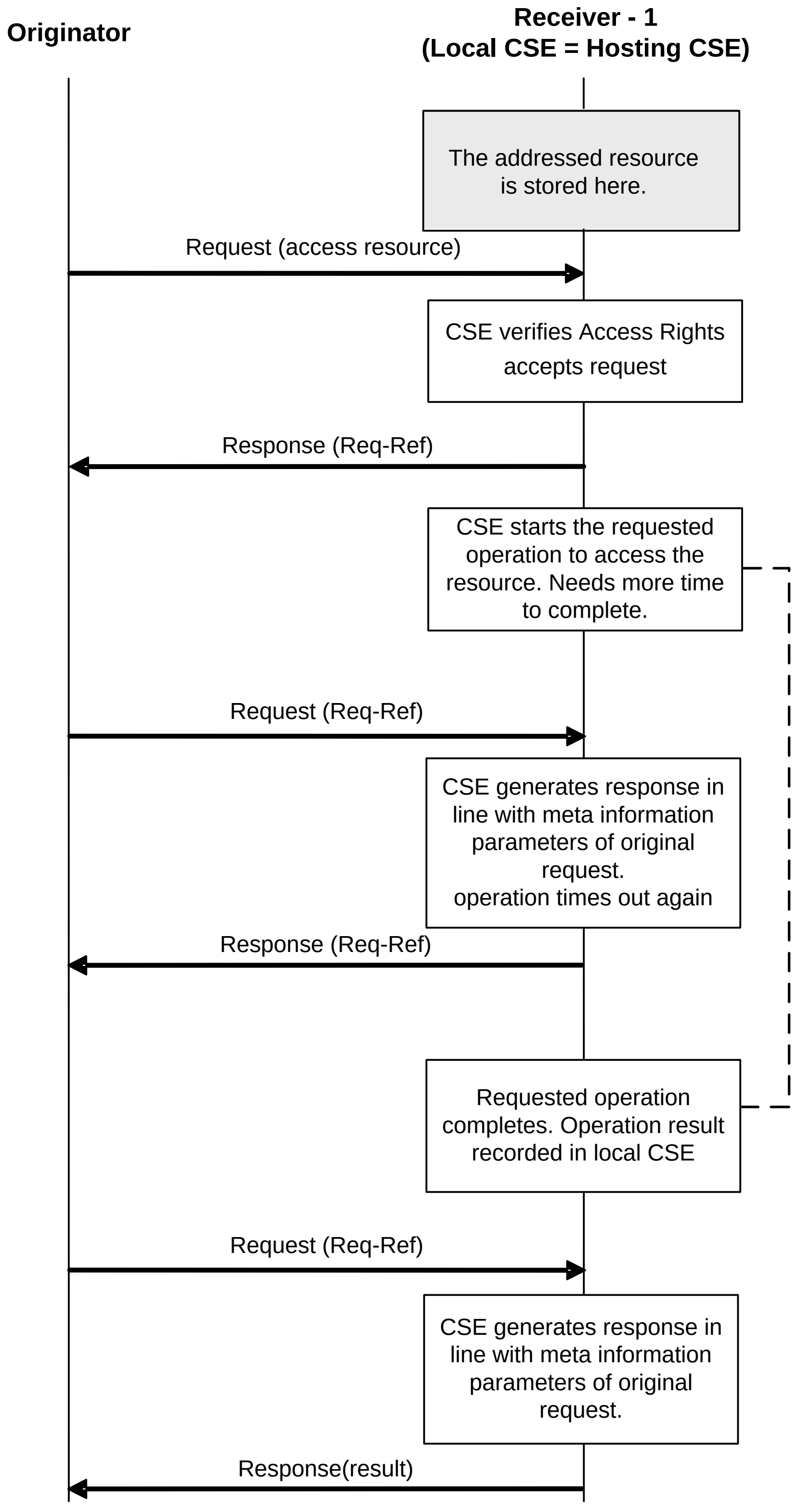

Originator
Receiver - 1
(Local CSE = Hosting CSE)
The addressed resource is stored here.
Request (access resource)
CSE verifies Access Rights accepts request
Response (Req-Ref)
CSE starts the requested operation to access the resource. Needs more time to complete.
Request (Req-Ref)
CSE generates response in line with meta information parameters of original request.
operation times out again
Response (Req-Ref)
Requested operation completes. Operation result recorded in local CSE
Request (Req-Ref)
CSE generates response in line with meta information parameters of original request.
Response(result)

Figure 8.2.2.2-2: Non-blocking access to resource in synchronous mode (Hosting CSE = Registrar CSE), requested operation completed after the second but before the third request

8.2.2.3 Asynchronous Case

In the asynchronous case, it is assumed that the Originator of a Request is able to receive notification messages, i.e. the Registrar CSE could send an unsolicited message to the Originator at an arbitrary time to send the result to a notification target. The possible mechanisms for the notification to reach the Originator are the same as in the case of a notification after a subscription.

In that case the information flow depicted in figure 8.2.2.3-1 is applicable. In this case it is assumed that the Originator of the Request provided a reference - notificationReference - for notification when the result of the requested operation is available.

Equivalent information flows are valid also for cases where the target resource of the requested operation is hosted on the Registrar CSE. From an Originator's perspective there is no difference as the later notification of the result of a requested operation would always be an exchange of request/response messages between the Originator and the Registrar CSE using reference to the original Request.

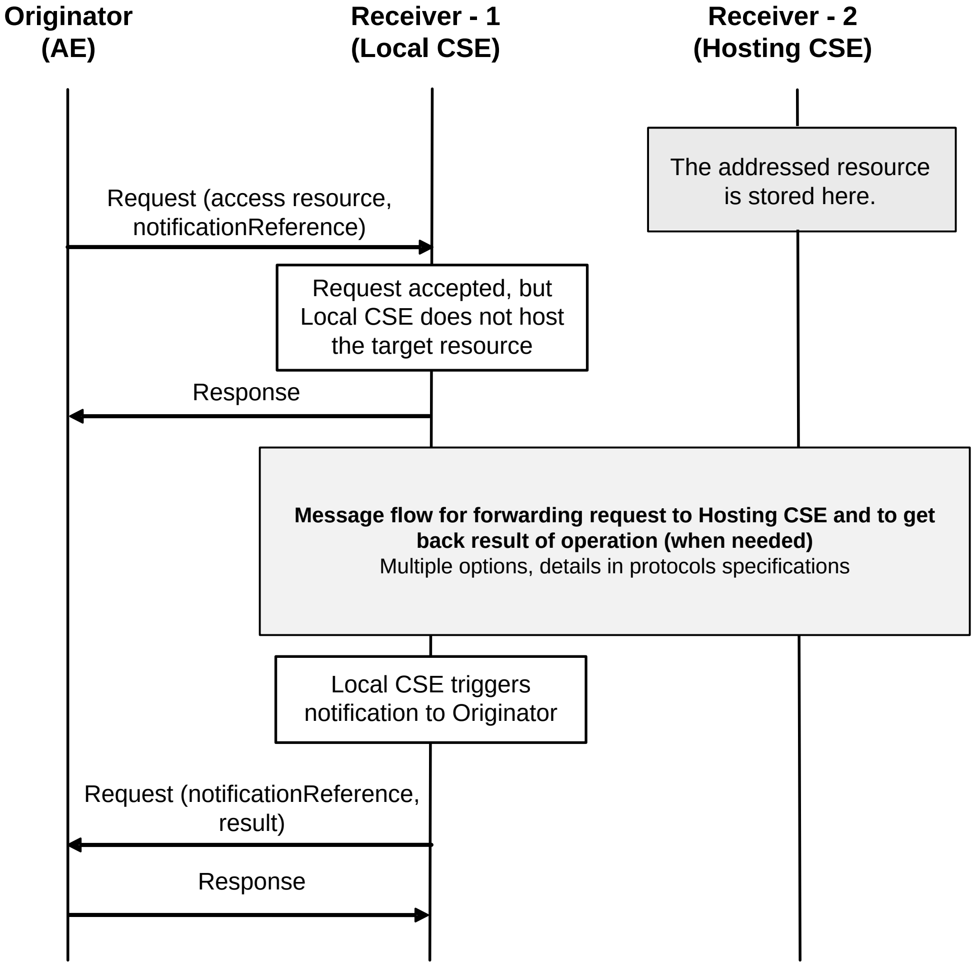

Originator (AE)
Receiver - 1 (Local CSE)
Receiver - 2 (Hosting CSE)
The addressed resource is stored here.
Request (access resource, notificationReference)
Request accepted, but Local CSE does not host the target resource
Response
Message flow for forwarding request to Hosting CSE and to get back result of operation (when needed)
Multiple options, details in protocols specifications
Local CSE triggers notification to Originator
Request (notificationReference, result)
Response

Figure 8.2.2.3-2: Non-blocking access to resource in notification mode (Hosting CSE not equal to Registrar CSE), Originator provided reference for notification

8.3 Description and Flows on Mcn Reference Point

Communications between the CSEs and the NSEs across the Mcn reference point and includes:

- The CSE(s) accessing network service functions provided by Underlying Networks; and
- Optimizing network service processing for Underlying Networks.

Such services normally are more than just the general transport services.

Communications which pass over the Mcn reference point to Underlying Networks include:

- Messaging services that are widely deployed by Applications and network operators using a number of existing mechanisms.
- Network APIs defined by other SDOs (e.g. OMA and GSMA) are used by network operators for their services.
- Interworking for services and security aspects for MTC (Machine Type Communications) has been defined by 3GPP.

Examples of service requests from a CSE towards the Underlying Networks are:

- Connection requests with/without QoS requirements.
- Payments, messages, location, bearer information, call control and other network capabilities (e.g. by using GSMA oneAPI, network APIs supporting protocols defined by other SDOs, or proprietary network APIs).
- Device triggering.
- Device management.
- Management information exchange such as charging/accounting records, monitoring and management data exchange.
- Location request.

8.4 Device Triggering

8.4.1 Definition and scope

Device Triggering is a means by which a node in the infrastructure domain (e.g. IN-CSE) sends information to a node in the field domain (e.g. ASN-CSE) to perform a specific task, e.g. to wake up the device, to establish communication from the field domain towards the infrastructure domain, or when IP address for the device is not available or reachable by the infrastructure domain.

Underlying Network functionality is used to perform device triggering for example, using alternate means of communication (e.g. SMS) with the Field Node.

NOTE: Device Triggering is applicable for the entities which are registered with IN-CSE.

Each Underlying Network type may provide different way of performing a device triggering, for example 3GPP has defined a dedicated interface for requesting device triggering. The normative references for applicable interfaces are as follows: TS 123 682 [i.17] and 3GPP2 X.S0068 [i.20]. Access specific mechanisms are covered in the annexes B and C.

8.4.2 General Procedure for Device Triggering

This clause covers different scenarios for device triggering.

8.4.2.1 Triggering procedure for targeting ASN/MN-CSE

This case describes the scenario where IN-CSE targets an ASN/MN-CSE (which is registered with the IN-CSE) for the Device Triggering request.

Figure 8.4.2 1-1 shows the general procedure for Device Triggering and, if required, for establishment of connectivity between IN-CSE and the Field Node.

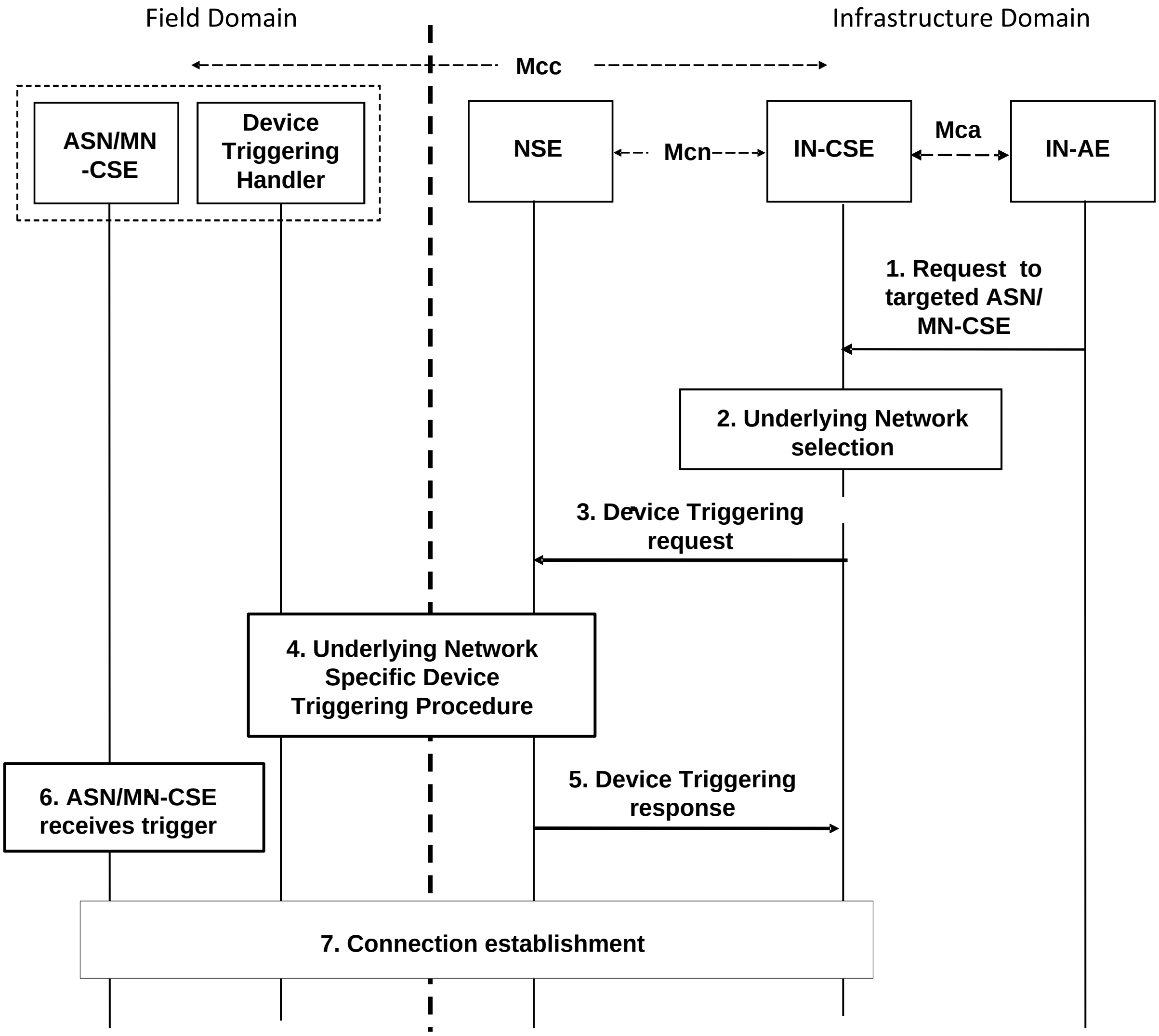

Field Domain
Infrastructure Domain
Mcc
ASN/MN -CSE
Device Triggering Handler
NSE
Mcn
IN-CSE
Mca
IN-AE
1. Request to targeted ASN/ MN-CSE
2. Underlying Network selection
3. Device Triggering request
4. Underlying Network Specific Device Triggering Procedure
5. Device Triggering response
6. ASN/MN-CSE receives trigger
7. Connection establishment

Figure 8.4.2 1-1: Device Triggering general procedure for CSE

NOTE 1: The IN and ASN/MN are assumed to be connected through the same Underlying Network.

NOTE 2: The Device Triggering Handler is a functional entity that receives the device triggering request, and it is dependent on the Underlying Network. The Device Triggering Handler is out of scope of this specification.

Pre-condition

The CSE which is the target of the device triggering has to be registered with the IN-CSE.

The CSE-PoA for the ASN/MN-CSE already contains either an IP address or none.

[optional] Step-1: Request to targeted ASN/MN-CSE

The IN-AE requests to perform one of the CRUD operations on a resource residing on the ASN/MN-CSE, the request is sent via the Mca reference point to the IN-CSE.

Step-2: Underlying network selection

The IN-CSE selects the Underlying Network and the mechanism to deliver the triggering request to the Underlying Network according to the configuration for connected Underlying Networks.

For example for 3GPP access network IN-CSE can use Tsp, Tsms and GSMA OneAPI, but the preferred mechanism is Tsp.

Step-3: Device Triggering request

IN-CSE issues the device triggering request to the selected Underlying Network.

NOTE 1: The Underlying Network dependent Device Triggering procedure for 3GPP and 3GPP2 systems are described in annexes B and Annex C respectively.

Some information provided to the selected Underlying Network for performing device triggering includes:

- M2M-Ext-ID associated with the ASN/MN-CSE as the target of the triggering request (see clause 7.1.8).
- Trigger-Recipient-ID associated with the ASN/MN-CSE (see clause 7.1.10). For example when 3GPP Underlying Network is used this identifier could map to Application-Port-ID.
- IN-CSE ID which could be used by the Underlying Network to authorize the IN-CSE for device triggering.

NOTE 2: The above Trigger-Recipient-ID is sent at registration.

[8.4.2.1.a] Editor's Note: M2M-Ext-ID is set on registration, how the IN-CSE gets it is FFS.

Step-4: Underlying Network Specific Device Triggering procedure

Device Triggering processing procedure is performed between the Underlying Network and the target Node which hosts the ASN/MN-CSE.

Step-5: Device Triggering response

The IN-CSE receives a response for the Device Triggering request via the Mcn reference point.

Step-6: ASN/MN-CSE Receives Device Trigger

[optional] Step-7 Connection establishment

In case that it is required by the Device Triggering request, connectivity is established between the ASN/MN-CSE and the IN-CSE and the renewal of the CSE-PoA might be needed.

8.5 Location Request

8.5.1 Definition and Scope

Location Request is a means by which a CSE requests the geographical or physical location information of a target CSE or AE hosted in a M2M Node to the location server located in the Underlying Network over Mcn reference point. This clause describes only the case of location request when the attribute locationSource is set to Network Based.

8.5.2 General Procedure for Location Request

This procedure describes a scenario wherein an AE or a CSE sends a request to obtain the location information of a target AE or CSE hosted in an M2M Node to the location server NSE, and the location server responses to the CSE with location information.

Figure 8.5.2.1-1 shows the general procedure for Location Request.

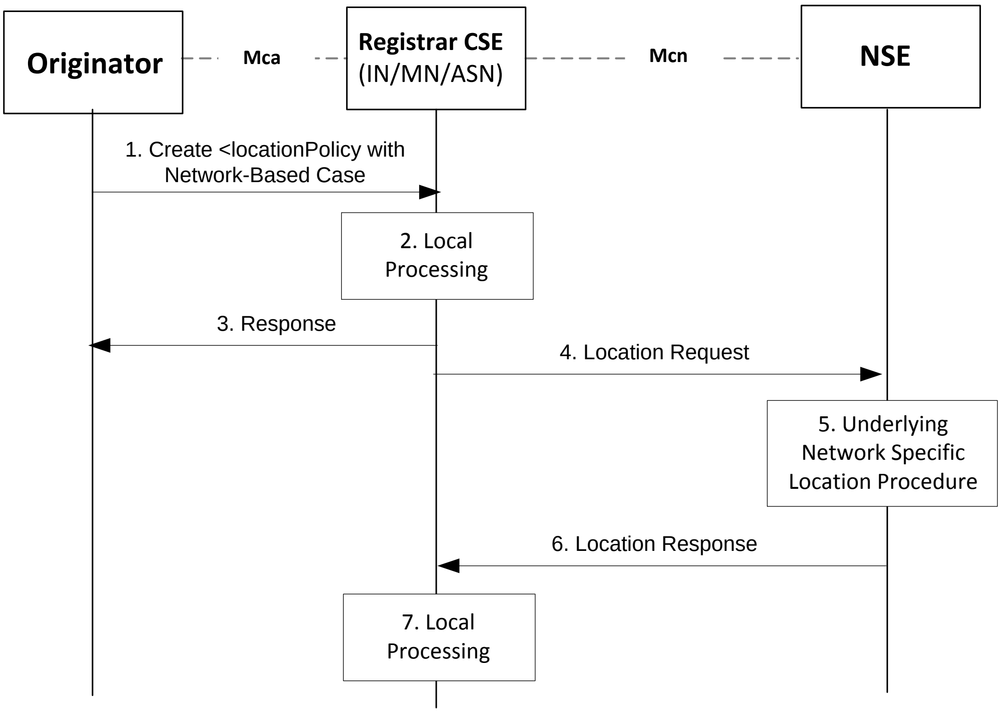

Originator
Mca
Registrar CSE (IN/MN/ASN)
Mcn
NSE
1. Create <locationPolicy with Network-Based Case
2. Local Processing
3. Response
4. Location Request
5. Underlying Network Specific Location Procedure
6. Location Response
7. Local Processing

Figure 8.5.2 1-1: General Procedure for Location Request

NOTE 1: Detailed descriptions for Step-1 to the Step-3 are described in the clause 10.2.11.1.

Step-1: Create <locationPolicy>

The Originator requests to CREATE <locationPolicy> resource at the Registrar CSE. The *locationSource* attribute of the <locationPolicy> resource shall be set to 'Network-Based' and the value for *locationTargetID* and *locationServer* attributes shall be set properly set for the Location Request.

Step-2: Local Processing for creating <locationPolicy> resource

After verifying the privileges and the given attributes, the Registrar CSE shall create the <locationPolicy> resource. Linked <container> resource can be created after successful creation of <locationPolicy> resource.

Step-3: Response for creating <locationPolicy>

The Registrar CSE shall respond with a Response message.

Step-4: Location Request

The Registrar CSE issues Location Request to the selected Underlying Network. For doing this, the local CSE shall transform the location configuration information received from the Originator into Location Request that is acceptable for the Underlying Network. For example, the Location Request can be one of existing location acquisition protocols such as OMA Mobile Location Protocol [i.5i.7] or OMA RESTful NetAPI for Terminal Location [i.8]. Additionally, the Registrar CSE shall provide default values for other parameters (e.g. required quality of position) in the Location Request according to local policies.

NOTE 2: The Location Request can be triggered by the given conditions, e.g.:

1) when the *locationUpdatePeriod* attribute has expired, or if the *locationUpdatePeriod* attribute is not given from the Step-1;

2) the <locationPolicy> is created; or

3) the linked <container> has been retrieved.

Step-5: Performing Location Procedure

The Underlying Network specific procedures are performed. This may involve getting location information from the target device or the network node. These procedures are outside the scope of oneM2M specifications.

Step-6: Location Response

The NSE responds to the local CSE with location information if the local CSE is authorized. If not, the NSE sends an error code back to the local CSE.

Step-7: Local Processing after Location Response

The received response shall be contained in the <container> resource that is related the <locationPolicy> resource.

NOTE 3: Please see the clause 10.2.11.2 for detail information.

NOTE 4: For notification regarding the location response towards the Originator, the subscription mechanism is used.

8.6 Connection Request

Connection request service is not defined in the present document.

8.7 Device Management

See clause 6.2.4 for a detailed description on the interaction with a Device Management

9 Resource Management

All entities in the oneM2M System, such as AEs, CSEs, data, etc. are represented as resources. A resource structure is specified as a representation of such resources. Such resources are uniquely addressable. Procedures for accessing such resources are also specified.

9.1 General Principles

The following are the general principles for the design of the resource model.

- The "type" of each resource shall be specified. New resource types shall be supported as the need for them is identified.
- The root of the resource structure in a CSE shall be assigned an absolute address. See clause 9.3.1 for additional information.
- The attributes for all resource type shall be specified.
- Each resource type can have multiple instances.
- All resources and associated attributes shall be uniquely addressable via their associated Universal Resource Identifiers (URI), consisting of:
  - a hierarchical URI based on the chain of child-parent relations;
  - a non-hierarchical URI made of a unique identifier addressable via the *CSEBase*.
  - Examples:
    - Root example "myCSE" (well known, common to hierarchical and non-hierarchical).

EXAMPLE 1: The following two examples address the same resource:

- //myCSE/123(hierarchical).
- //myCSE/234 (non-hierarchical).

EXAMPLE 2: The following two examples address the same resource:

- //myCSE/myApplication/987 (hierarchical).
- //myCSE/234 (non-hierarchical).

- The non-hierarchical URI is always stored in the *parentID* attribute. That is a mandatory attribute.
- In case of hierarchical URI, the relationship parent-child and vice-versa is determined in the hierarchical URI. When created via the non-hierarchical URI, the parent child-relation is determined by the *parentID* attribute.
- Both hierarchical and non-hierarchical URIs shall be supported by all CSEs.

9.2 Resources

This clause introduces the resources used in a CSE. A resource scheme is used for modelling the resource structure and associated relationships. Clause 9.5 provides guidelines on how to describe a resource. This document identifies three categories of resources:

- Normal resources (clause 9.2.1).
- Virtual resources (clause 9.2.2).
- Announced resources (clause 9.2.3).

9.2.1 Normal Resources

Normal resources include the complete set of representations of data which constitutes the base of the information to be managed.

Unless qualified as either "virtual" or "announced", the resource types in this document are normal resources.

9.2.2 Virtual Resources and Attributes

A virtual resource or a virtual attribute is used to trigger processing and/or retrieve results, but they do not have a permanent representation in a CSE.

9.2.3 Announced Resources

An announced resource is a resource at a remote CSE that is linked to the original resource that has been announced, and it maintains some of the characteristics of the original resource.

Resource announcement can facilitate resource discovery. The announced resource at a remote CSE can also be used for creating child resources at the remote CSE that are not present as children of the original resource or are not announced children of the original resource.

The following are the resource specification guidelines for resource announcement:

- In order to support announcement of resources, an additional column in the resource template (clause 9.5.1), shall specify the attributes to be announced for inclusion in the associated announced resource type.
- For each announced <*resourceType*>, the addition of suffix "Annc" to the original <*resourceType*> shall be used to indicate its associated announced resource type. For example, resource <*containerAnnc*> shall indicate the announced resource type for <*container*> resource; <*groupAnnc*> shall indicate announced resource type for <*group*> resource, etc.

9.3 Resource Addressing

9.3.1 Generic Principles

There are three different methods for addressing a resource within the oneM2M resource structure. They are as follows:

- **Hierarchical URI Method:** The resource can be addressed by a URI, over Mca, Mcc, and Mcc' reference points, with the actual path portion of the URI defining the entire relationship for the target resource within the resource structure. This is a structured representation of the resources within a CSE where the parent relationship chain is embedded in the resource address.

EXAMPLE 1: IN-CSEID.m2m.myoperator.org/CSERoot/myAppX/myContainerY.

- **Non-Hierarchical URI Method:** The resource can be addressed by a URI, over Mca, Mcc, and Mcc' reference points, with the resource identifier given by its hosting CSE during the resource creation procedure. Hence the actual parent relationship chain is not known a priori and the hosting CSE needs to resolve the logical location of the target resource in the chain of relationship within the resource structure.

EXAMPLE 2: "IN-CSEID.m2m.myoperator.org/CSERoot/mCY" where the same container of the previous example is directly addressed. *mCY* is the resource identifier that was given by the hosting CSE for *myContainerY*.

- **ID Base Method:** The resource can be addressed via two parameters over Mca reference point; namely the CSE-ID of the host where the resource is located, and/or the resource identifier of the actual target resource. This case is applicable only to intra-domain routing. To proxy this request onwards over the Mcc reference point, if applicable, the registrar CSE can proxy the request to the IN CSE including these two parameters; optionally the registrar CSE shall use the CSE-ID to generate the complete target host name in accordance with the rules define in clause 6.4.1.1.

Note in this scenario it is assumed that the actual node identifiers are used in the host name to enable the generation of the host name. The CSEBase is used to identify the CSE-ID.

This method is an optimization of the second method, since the host name included in the URI has to be generated, as an option, in this case by the registrar CSE before it can send the request to the target CSE (via the IN node).

EXAMPLE 3: The same resource addressed in the previous example will be provided by the requesting entity to the receiving CSE by means of the identifiers "CSE4-CSEID" and "mCY" and then it will be resolved to "CSE4-CSEID.IN-CSEID.m2m.myoperator.org/CSERoot/mCY".

These 3 methods shall all be supported by all M2M nodes, notably the registrar CSEs receiving requests, before they proxy these requests any further, where applicable.

The CSEBase, which is the first element in the path portion of the URI, allows to easily distinguish different CSEs on the same IP host.

9.3.2 Resource addressing via a FQDN URI

Resources shall be uniquely addressable via FQDN URI as one of the methods in clause 9.3.1. The following are examples of addressable resources.

- /<CSEBase>/Entity Instance_1.
- /<CSEBase>/Entity Instance_n.
- /<CSEBase>/Application Instance_1.
- /<CSEBase>/Application Instance_n.
- /<CSEBase>/Container_1/Container_2.

<CSEBase> is the root for addressing all other resources it contains. The *<CSEBase>* shall represent the first element of the path representing a resource in an absolute URI, as per [i.12].

The authority part of the absolute URI, as per [i.14] and [i.12] shall be the FQDN or IP address for the CSE hosting the resources and shall conform to the naming convention described in clause 6.4.1.1.

9.4 Resource Structure

9.4.1 Relationships between Resources

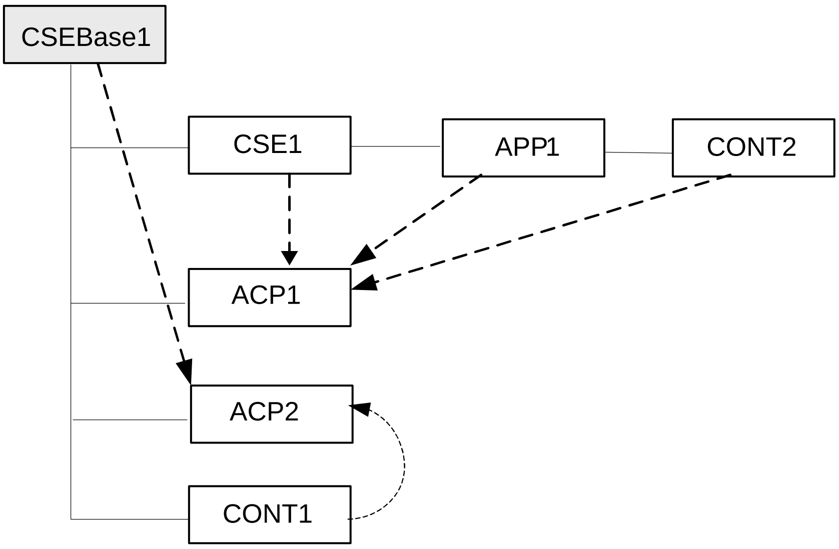


Figure 9.4.1-1: Resource Relationships Example in a CSE

NOTE: The resources shown in the above figure are:

- CSEBase1 is the name of a resource of type <*CSEBase*>
- CSE1 is the name of a resource of type <*remoteCSE*>
- APP1 is the name of a resource of type <*AE*>
- CONT1 and CONT2 are the names of resources of type <*container*>
- ACP1 and ACP2 are the names of resources of type <*accessControlPolicy*>

The solid line in Figure 9.4.1-1 represents parent-child relation, which is supported by a link (e.g. *parentID*) in the non-hierarchical addressing scheme, and by the URI scheme in the hierarchical addressing scheme.

Dashed line in Figure 9.4.1-1 represents a link i.e. a relationship between the resources (e.g. relationship between the APP1 resource and the ACP1).

Figure 9.4.1-1 provides an example of a resource structure. The represented resources can be addressed by using one of the methods described in clause 9.3.1. Resources in the oneM2M System are linked with each other and they respect the containment relationship. The methods for linking resources are described in clause 9.4.2.

A link shall contain the following information:

- *linkedResourceURI:* The target linked resource is given by using the URI of that resource.
- *linkRelation:* Describes the relationship that the current resource has with the linked resource (only in one direction, i.e. from this resource to the linked resource).

9.4.2 Link Relations

The following link relations are defined.

Table 9.4.2-1: Link Relations

Linked Resource Type (link destination)	Linking Resource Types (link origin)	Linking Method	Description
accessControPolicy	Several (e.g., *node, AE, remoteCSE, container*)	Attribute named *accessControlPolicyIDs*	See clause 9.6.2
node	*CSEBase, remoteCSE, AE*	Attribute named *nodeLink*	See clause 9.6.18
CSEBase or *remoteCSE*	*node*	Attribute named *hostedCSEID* OR parent resource of type *CSEBase*	See clause 9.6.3 See clause 9.6.4
a parent resource of any resourceType	a child resource of any resourceType	Attribute named *parentID*	See clause 9.6.1
a child resource of any resourceType	a parent resource of any resourceType	Child resource of a specific type	See clause 9.6.5
mgmtObj	*mgmtObj*	Attribute named: *mgmtLink*	See clause 9.6.15

9.5 Resource Type Specification Conventions

The following conventions are used for the specification of resources.

Resources are specified via a tabular notation and the associated graphical representation as follows:

- The resources are specified in association with a CSE. The resources are the representation in the CSE of the components and elements within the oneM2M System. Other CSEs, AEs, application data representing sensors, commands, etc. are known to the CSE by means of their resource representation. Resource, Child Resource and Attributes are defined in clause 3.1 and are restated below for readability.
    - **Resource:** A Resource is a uniquely addressable entity in oneM2M architecture. A resource is transferred and manipulated using CRUD operations (see clause 10.1). A resource can contain child resource(s) and attribute(s).
    - **Child Resource:** It is a sub-resource of another resource that is its parent resource. The parent resource contains references to the child resources(s).
    - **Attribute:** Stores information pertaining to the resource itself.
- The set of attributes, which are common to all resources, are not detailed in the graphical representation of a resource.
- Resource names and attribute names are strings in lower case. In case of a composed name, the subsequent word(s) start with a capital letter; e.g. *accessControlPolicy*, *creationTime*, *expirationTime*.
- Resource type names and attribute names are written in *italic* form in this document.
- A string delimited with '<' and '>' e.g. *<resourceType>* is a placeholder for the type of a resource.
- A string delimited with '[' and ']' e.g. *[resourceName]* is a placeholder for the name of a resource or an attribute.

The resources are specified as shown below.

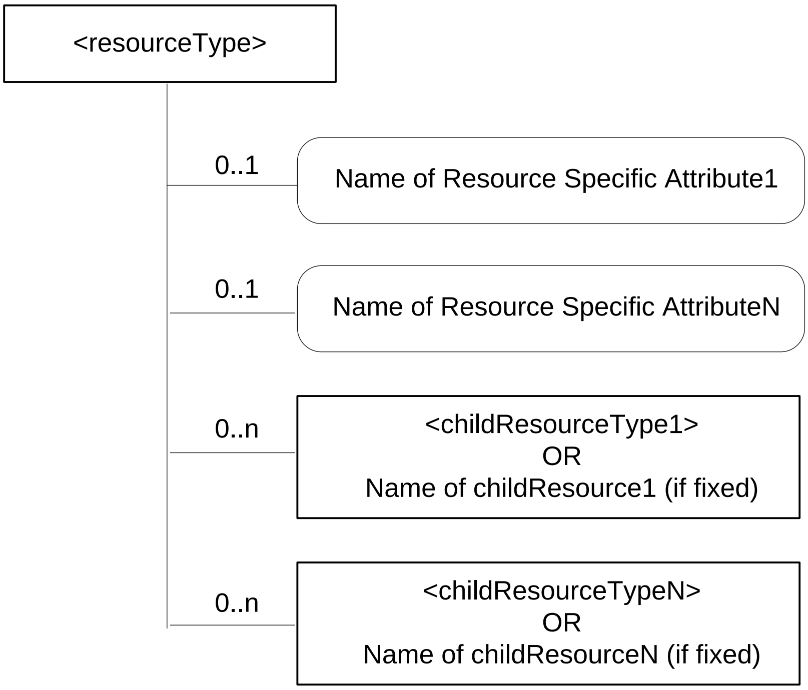

<resourceType>
0..1
Name of Resource Specific Attribute1
0..1
Name of Resource Specific AttributeN
0..n
<childResourceType1>
OR
Name of childResource1 (if fixed)
0..n
<childResourceTypeN>
OR
Name of childResourceN (if fixed)

Figure 9.5-1: *<resourceType>* representation convention

The resource specification provides the graphical representation for the resource as the figure 9.5-1. The graphical representation of a resource shows the multiplicity of the attributes and child resources. The set of attributes, which are common to all resources are not detailed in the graphical representation of a resource. The following graphical representations are used for representing the attributes and child resources:

- Square boxes are used for the resources;
- Square boxes with round corners are used for attributes.

Child resources in a *<resourceType>* are detailed as shown in table 9.5-1.

The child resource table for an announce-able <resourceType> resource includes an additional column titled '*<resourceTypeAnnc>* Child Resource Types', indicating the type of announced resources. See the clause 9.6.25 for further details.

An announced resource may have child resources, and such child resources can be of type normal or announced. When a resource is announced, the associated child resources are announced independently of the original resource, as needed by the resource announcing CSE. The child resources at the announced resource are of the child resource Announced type.

When child resources at the announced resource are created locally by the remote CSE, the child resources are of normal child resource type.

Table 9.5-1: Child Resources of *<resourceType>*

Child Resources of *<resourceType>*	Child Resource Type	Multiplicity	Description	*<resourceTypeAnnc>* Child Resource Types
<Fill in the name of Child Resource1 if a fixed name is required or [variable] if no fixed name is required>	<Fill in the type of Child Resource1>	<Fill in Multiplicity>	See clause <XRef> <clause> where the type of this child resource is described.	<Fill the child resource type for the announced resource. It can be none or <crTypeAnnc> or <crType>; where the <crType> is the child resource type of the original Child Resource1.
<Fill in the name of Child ResourceN if a fixed name is required or [variable] if no fixed name is required>	<Fill in the type of Child ResourceN>	<Fill in Multiplicity>	See clause <XRef> <clause> where the type of this child resource is described.	<Fill the child resource type for the announced resource. It can be none or <crTypeAnnc> or <crType>; where the <crType> is the child resource type of the original Child ResourceN.

Attributes in a *<resourceType>* are detailed as shown in table 9.5-2

The attributes table for announce-able *<resourceType>* resource includes an additional column titled 'Attributes for <resourceTypeAnnc>', indicating the attributes that are to be announced for that <resourceType>. See the clause 9.6.25 for further details.

Table 9.5-2: Attributes of *<resourceType>* resource

Attributes of *<resourceType>*	Multiplicity	RW/ RO/ WO	Description	*<resourceTypeAnnc>* (MA/OA/NA)
<Fill in name of Common Attribute1>	<Fill in Multiplicity>	<Fill in RW or RO or WO>	Provide description of this attribute - to be moved later to a common attribute clause.	<Fill in MA or OA or NA>
<Fill in name of Common AttributeN>	<Fill in Multiplicity>	<Fill in RW or RO or WO>	Provide description of this attribute - to be moved later to a common attribute clause.	<Fill in MA or OA or NA>
<Fill in name of Resource Specific Attribute1>	<Fill in Multiplicity>	<Fill in RW or RO or WO>	Provide description of this attribute - to be moved later to a central attribute table that also defines the type of the attribute, allowed ranges etc.	<Fill in MA or OA or NA>
<Fill in name of Resource-Specific AttributeN>	<Fill in Multiplicity>	<Fill in RW or RO or WO>	Provide description of this attribute - to be moved later to a central attribute table that also defines the type of the attribute, allowed ranges etc.	<Fill in MA or OA or NA>

In case of misalignment of the graphical representation of a resource and the associated tabular representation, tabular representation shall take precedence.

The access modes for *attributes* can assume the following values:

- Read/Write (RW): all operations are allowed for the attribute (Create/Update/Retrieve/Delete/Notify).
- Read Only (RO): the value of the attribute is set by the hosting CSE when the resource is Created. Such an attribute can only be read.
- Write Once (WO): the value of the attribute is set when the resource is Created based on information from the originator. Such an attribute can then only be read.

The multiplicity, both for the child resources and the attributes can have the following values:

- A value of "0" indicates that the child resource/attribute is not present.
- A value of "1" indicates that the child resource/attribute is present.
- A value of "0..1" indicates that the child resource/attribute could not be present. If present, it can have an instance of one only.
- A value of "0..n" indicates that the child resource could not be present. If present, multiple instances are supported.
- A value of "1..n" indicates that the child resource is always present. It has at least one instance and can have multiple instances.
- An attribute multiplicity post-fixed with (L) indicates that it is a list of values.

The attributes for *<resourceTypeAnnc>* in the attribute table can have the following set of values:

- **MA** (Mandatory Announced): Such attributes in the original resource are announced to the announced resources. The content of such announced attributes is the same as the content of the original resource.
- **OA** (Optional Announced): Such attributes in the original resource may be announced to the announced resources depending on the contents of the *announcedAttribute* attribute at the original resource. The content of such announced attributes is of the same as the content of the original attributes.
- **NA** (Not Announced): Such attributes are not announced to the announced resources.

9.5.1 Handling of Unsupported Resources/Attributes/Sub-resources within the M2M System

Any CSE shall respond to a received request targeted to it and that includes resource(s), resource attribute(s) or sub-resource(s) that are unsupported by the target CSE, by sending an appropriate error code back to the request originator.

When a CSE is not the target entity of a received request that includes resource(s), resource attribute(s) or sub-resource(s) that are unsupported by the CSE, the CSE shall attempt to forward the received request to the targeted entity. If the CSE cannot forward the received request for any reason, the forwarding CSE shall respond to the received request by sending an appropriate error code back to the request originator. This specification includes both mandatory and optional functionalities for interfaces between oneM2M entities. Thus, the functionality implemented for the interfaces may not include all the functionalities specified in this document.

9.6 Resource Types

Table 9.6-1 introduces the normal and virtual resource types and their related child or parent resource types. Details of each resource type follow in the remainder of this clause.

Table 9.6-1 lists the supported <*resourceTypes*>. An addition of suffix "Annc" to such <*resourceTypes*> indicates the associated announced resource type.

Table 9.6-1 Resource Summary

Resource Type	Short Description	Child Resource Types	Parent Resource Types	Clause
activeCmdhPolicy	Provides a link to the currently active set of CMDH policies.	*None*	*CSEBase*	D.12.1
accessControlPolicy	Stores a representation of privileges. It is associated with resources that shall be accessible to entities external to the hosting CSE. It controls "who" is allowed to do "what" and the context in which it can be used for accessing the resources	*subscription*	*AE, remoteCSE, CSEBase*	9.6.2
contentInstance	Represents a data instance in the container resource	*subscription*	*container*	9.6.7
AE	Stores information about the AE. It is created as a result of successful registration of an AE with the registrar CSE	*subscription, container, group, accessControlPolicy, mgmtObj, commCapabilities, pollingChannel*	*remoteCSE, CSEBase*	9.6.5
cmdhBuffer	Defines CMDH buffer usage limits	*subscription*	*cmdhPolicy*	D.12.8
cmdhDefaults	Defines CMDH default values	*cmdhDefEcValue, cmdhEcDefParamValues subscription*	*cmdhPolicy*	D.12.2
cmdhEcDefParamValues	Represent a specific set of default values for the CMDH related parameters	*subscription*	*cmdhDefaults*	D.12.4
cmdhDefEcValue	Defines a value for the **ec** (event category) parameter of an incoming request when it is not defined	*subscription*	*cmdhDefaults*	D.12.3
cmdhLimits	Defines limits for CMDH related parameter values	*subscription*	*cmdhPolicy*	D.12.5
cmdhNetworkAccessRules	Defines rules for the usage of underlying networks	*cmdhNwAccessRule subscription*	*cmdhPolicy*	D.12.6
cmdhNwAccessRule	Defines a rule for the usage of underlying networks	*schedule subscription*	*cmdhNetworkAccessRules*	D.12.7
cmdhPolicy	A set of rules defining which CMDH parameters will be used by default	*cmdhDefaults, cmdhLimits, cmdhNetworkAccessRules, cmdhBuffer subscription*	*CSEBase*	D.12
container	Shares data instances among entities. Used as a mediator that takes care of buffering the data to exchange "data" between AEs and/or CSEs. The exchange of data between AEs (e.g. an AE on a Node in a field domain and the peer-AE on the infrastructure domain) is abstracted from the need to set up direct connections and allows for scenarios where both entities in the exchange do not have the same reachability schedule	*container, contentInstance, subscription,*	*application, container, remoteCSE, CSEBase*	9.6.6

Resource Type	Short Description	Child Resource Types	Parent Resource Types	Clause
CSEBase	The structural root for all the resources that are residing on a CSE. It shall store information about the CSE itself	*remoteCSE, node, application, container, group, accessControlPolicy, subscription, mgmtObj, mgmtCmd, locationPolicy, statsConfig*	*None*	9.6.3
delivery	Forwards requests from CSE to CSE	*subscription*	*CSEBase*	9.6.11
eventConfig	Defines events that trigger statistics collection	*subscription*	*statsConfig*	9.6.23
execInstance	The Execution Instance resource contains all execution instances of the same management command *mgmtCmd*	*subscription*	*mgmtCmd*	9.6.17
group	Stores information about resources of the same type that need to be addressed as a Group. Operations addressed to a Group resource shall be executed in a bulk mode for all members belonging to the Group	*fanOutPoint* *subscription*	*Application, remoteCSE, CSEBase*	9.6.13
locationPolicy	Includes information to obtain and manage geographical location. It is only referred from container, the *contentInstances* of the container provides location information	*subscription*	*CSEBase*	9.6.10
fanOutPoint	Virtual resource containing target for group requests It is used for addressing bulk operations to all the resources that belong to a group	*None*	*group*	9.6.14
mgmtCmd	Management Command resource represents a method to execute management procedures required by existing management protocols	*execInstance* *subscription*	*CSEBase*	9.6.16
mgmtObj	Management Object resource represents management functions that provides an abstraction to be mapped to external management technology. It represents the node and the software installed in the node	*parameters* *subscription*	*remoteCSE, CSEBase*	9.6.15 Annex D
m2mServiceSubscription	Data pertaining to the M2M Service Subscription	*nodeInfo* *subscription*	*None documented*	9.6.19
node	Represents specific Node information	*schedule, mgmtObj* *subscription*	*CSEBase, remoteCSE*	9.6.18
nodeInfo	Node information	*subscription*	*m2mServiceSubscription*	9.6.20
parameters	Provides a mechanism to describe the management object in a generic way to easily import information from existing management protocols	*parameters* *subscription*	*mgmtObj, parameters*	9.6.16
pollingChannel	Represent a channel that can be used for a request-unreachable entity	*None*	*remoteCSE, application*	9.6.21

Resource Type	Short Description	Child Resource Types	Parent Resource Types	Clause
remoteCSE	Represents a remote CSE for which there has been a registration procedure with the registrar CSE identified by the CSEBase resource	*application, container, group, accessControlPolicy, subscription, mgmtObj, pollingChannel, node*	*CSEBase*	9.6.4
request	Expresses/access context of an issued Request	*subscription*	*CSEBase*	9.6.12
schedule	Contains scheduling information for delivery of messages	*subscription*	*node, subscription, cmdhNwAccessRule*	9.6.39
statsCollect	Defines triggers for the IN-CSE to collect statistics for applications	*subscription*	*CSEBase (in IN-CSE)*	9.6.24
statsConfig	Stores configuration of statistics for applications	*eventConfig* *subscription*	*CSEBase (in IN-CSE)*	9.6.22
subscription	Subscription resource represents the subscription information related to a resource. Such a resource shall be a child resource for the subscribe-to resource	*schedule*	*accessControlPolicy, application, cmdhBuffer, cmdhDefaults, cmdhEcDefParamValues, cmdhDefEcValue, cmdhLimits, cmdhNetworkAccessRules, cmdhNwAccessRule, cmdhPolicy, container, CSEBase, delivery, eventConfig, execInstance, group, contentInstance, locationPolicy, mgmtCmd, mgmtObj, m2mServiceSubscription, node, nodeInfo, parameters, remoteCSE, request, schedule, statsCollect, statsConfig*	9.6.8

9.6.1 Common Attributes

Many of the attributes of the resources described in the present document are common. Such attributes are described here once in order to avoid duplicating the description for every resource that contains it.

Attributes that are only used in one or two resource types are described only in the clause specific for that resource type.

Table 9.6.1-1: Common Attributes

Common Attribute	Description
resourceType	Resource Type. This Write Once (assigned at creation time. and then cannot be changed) resourceType attribute identifies the type of resources as specified in clause 9.6. Each resource shall have a *resourceType* attribute.
resourceID	This attribute is an identifier for resource that is used for 'non-hierarchical URI method' or 'IDs based method' cases. This attribute shall be provided by the hosting CSE when it accepts a resource creation procedure. The hosting CSE shall assign a *resourceID* which is unique in the CSE.
parentID	The system shall assign the value to this attribute according to the parameters given in the CREATE Request. It establishes the parent-child relationship by identification of the parent of this child resource. Such identifier shall use the non-hierarchical URI representation. For example, an AE resource with the identifier "myAE1" which has been created under the resource "…//example.com/oneM2M/myCSE", the value of the *parentID* attribute will contain "…//parent*ID*".
accessControlPolicyIDs	The attribute contains a list of identifiers (either an ID or a URI depending if it is a local resource or not) of an <*accessControlPolicy*> resource. The privileges defined in the <*accessControlPolicy*> resource that are referenced determine who is allowed to access the resource containing this attribute for a specific purpose (e.g. Retrieve, Update, Delete, etc.). If a resource type does not have an *accessControlPolicyIDs* attribute definition, then the *accessControlPolicy* for that resource is governed in a different way, for example, the *accessControlPolicy* associated with the parent may apply to a child resource that does not have an *accessControlPolicyIDs* attribute definition, or the privileges for access are fixed by the system. Refer to the corresponding resourceType and procedures to see how permissions are handled in such cases. If a resource type does have an *accessControlPolicyIDs* attribute definition, but the (optional) *accessControlPolicyIDs* attribute is not set, or it is set to a value that does not correspond to a valid, existing <*accessControlPolicy*> resource, or it refers to an <*accessControlPolicy*> resource that is not reachable (e.g. because it is located on a remote CSE that is offline or not reachable), then the system default access permissions shall apply. All resources are accessible only if the privileges from the Access Control Policy grants it, therefore all resources shall have an associated *AccessControlPolicyIDs* attribute, either explicitly (setting the attribute in the resource itself) or implicitly (either by using the parent privileges or the system defaults). Which means that the system shall provide a default access privileges in case that the Originator does not provide a specific *AccessControlPolicyIDs* during the creation of the resource, Default access grants the configured privileges to the originator (e.g. depending on the prefix of URI of the resource). This attribute is absent from the resource in some cases, especially if the resource shall have the same privileges of the parent resource; such an attribute is therefore not needed. To update this attribute, a hosting CSE shall check whether an originator has Update permission in any *selfPrivileges* of the <*accessControlPolicy*> resources which this attribute originally indicates.
creationTime	Time/date of creation of the resource. This attribute is mandatory for all resources and the value is assigned by the system at the time when the resource is locally created. Such an attribute cannot be changed.

Common Attribute	Description
expirationTime	Time/date after which the resource will be deleted by the hosting CSE. This attribute can be provided by the originator, and in such a case it will be regarded as a hint to the hosting CSE on the lifetime of the resource. The hosting CSE can however decide on the real *expirationTime*. If the hosting CSE decides to change the *expirationTime* attribute value, this is communicated back to the originator. The lifetime of the resource can be extended by providing a new value for this attribute in an UPDATE operation. Or by deleting the attribute value, e.g. by not providing the attribute when doing a full UPDATE, in which case the hosting CSE can decide on a new value. This attribute shall be mandatory. If the originator does not provide a value in the CREATE operation the system shall assign an appropriate value depending on its local policies and/or M2M service subscription agreements.
lastModifiedTime	Last modification time/date of the resource. This attribute shall be mandatory and its value is assigned automatically by the system each time that the addressed target resource is modified by means of the UPDATE operation.
stateTag	An incremental counter of modification on the resource. When a resource is created, this counter is set to 0, and it will be incremented on every modification of the resource. NOTE 1: In order to enable detection of overflow, the counter needs to be capable of expressing sufficiently long numbers. . NOTE 2: This attribute has the scope to allow identifying changes in resources within a time interval that is lower than the one supported by the attribute *lastModifiedTime* (e.g. less than a second or millisecond). This attribute can also be used to avoid race conditions in case of competing modifications. Modifications (e.g. Update/Delete) can be made on the condition that this attribute has a given value.
labels	Tokens used as keys for discovering resources. This attribute is optional and if not present it means that the resource cannot be found by means of discovery procedure which uses *labels* as key parameter of the discovery.
link	This attribute shall be present only on the announced resource. This attribute shall provide the link (URI) to the original resource.
announceTo	This attribute may be included in a CREATE or UPDATE Request in which case it contains a list of URIs/CSE-IDs which the resource being created/updated shall be announced to. This attribute shall only be present at the original resource if it has been successfully announced to other CSEs. This attribute maintains the list of URIs to the successfully announced resources. Updates on this attribute will trigger new resource announcement or de-announcement.
announcedAttribute	This attributes shall only be present at the original resource if some Optional Announced **(OA)** type attributes have been announced to other CSEs. This attribute maintains the list of the announced Optional Attributes (**OA** type attributes) in the original resource. Updates to this attribute will trigger new attribute announcement if a new attribute is added or de-announcement if the existing attribute is removed.

9.6.2 Resource Type *accessControlPolicy*

The <*accessControlPolicy*> resource is comprised of *privileges* and *selfPrivileges* attributes which represent a set of access control rules defining which entities (defined as *accessControlOriginators*) have the privilege to perform certain operations (defined as *accessContolOperations*) within specified contexts (defined as *accessControlContexts*) and are used by the CSEs in making access decision to specific resources.

Each access control rule defines what is allowed. So for sets of Access Control rules an operation is permitted if it is permitted by some/any of the access control rules in the set. As a consequence, a combination of access control rules will never be conflicting.

For a resource that is not of <*accessControlPolicy*> resource type, the common attribute *accessControlPolicyIDs* for such resources (defined in table 9.6.1-1) contains a list of identifiers which link that resource to <*accessControlPolicy*> resources. The CSE access decision for such a resource shall follow the evaluation of the set of access control rules expressed by the *privileges* attributes defined in the identified <*accessControlPolicy*> resources.

The *selfPrivileges* attribute represents the set of access control rules for the resource <*accessControlPolicy*> itself.

The CSE access decision for <*accessControlPolicy*> resource shall follow the evaluation of the set of access control rules expressed by the *selfPrivileges* attributes defined in the identified <*accessControlPolicy*> resource itself.

The Access Control Policies (ACPs)shall be used by the CSE to control access to the resources as specified in this document and in TS-0003 [i-3].

The ACP is designed to fit different access control models, such as access control lists, role or attribute based access control.

The ACP associates *accessControlOriginators* with the privilege to perform certain operation within a given context, that maps in the definition of a role. Multiple ACPs can be associated to the same resource.

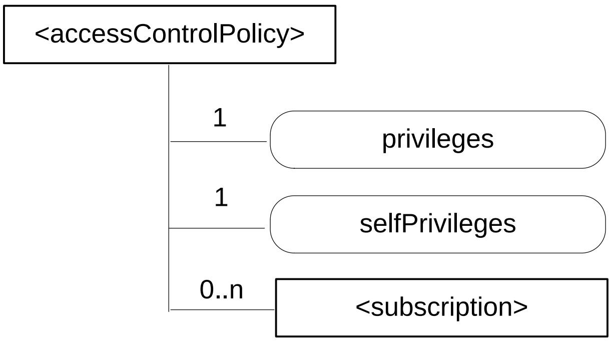


Figure 9.6.2-1: Structure of <*accessControlPolicy*> resource

The <*accessControlPolicy*> resource shall contain the child resource specified in table 9.6.2-1.

Table 9.6.2-1: Child resources of <*accessControlPolicy*> resource

Child Resources of <*accessControlPolicy*>	Child Resource Type	Multiplicity	Description	<*accessControlPolicyAnnc*> Child Resource Types
[variable]	*<subscription>*	0..n	See clause 9.6.8	*<subscription>*

The <*accessControlPolicy*> resource shall contain the attributes specified in table 9.6.2-2.

Table 9.6.2-2: Attributes of <*accessControlPolicy*> resource

Attributes of <*accessControlPolicy*>	Multiplicity	RW/ RO/ WO	Description	<*accessControlPolicyAnnc*> Attributes
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.	NA
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.	MA
parentID	1	RO	See clause 9.6.1 where this common attribute is described.	NA
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.	MA
labels	0..1	RW	See clause 9.6.1 where this common attribute is described.	MA
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.	NA
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.	NA
announceTo	1	RW	See clause 9.6.1 where this common attribute is described.	NA
announcedAttribute	1	RW	See clause 9.6.1 where this common attribute is described.	NA
privileges	1	RW	Represent a set of access control rules that applies to resources referencing this <*accessControlPolicy*> resource using the *accessControlPolicyID* attribute.	MA
selfPrivileges	1	RW	Represent the Set of access control rules that apply to the <*accessControlPolicy*> resource itself.	MA

The set of access control rules represented in *privileges* and *selfPrivileges* attributes are comprised of 3-tuples (*accessControlOriginators*, *accessControlContexts*, *accessControlOperations*) with parameters shown in table 9.6.2-3 which are further described in the following clauses.

If *privileges* attribute contains no 3-tuples then this represent an empty set of the access control rules.

The *selPrivileges* attribute shall contain at least one 3-tuples.

The CSE access granting mechanism shall follow the procedure described in TS-0003 [i-3] in clause 7.1 (Authorization and access control).

Table 9.6.2-3: Parameters in access-control-rule-tuples

Name	Description
accessControlOriginators	See 9.6.2.1
accessControlContexts	See 9.6.2.2
accessControlOperations	See 9.6.2.3

9.6.2.1 *accessControlOriginators*

The *accessControlOriginators* is a mandatory parameter in an access-control-rule-tuple. It represents the set of originators that shall be allowed to use this access control rule. The set of originators is described as a list of parameters, where the types of the parameter can vary within the list. Table 9.6.2.1-1 describes the supported types of parameters in *accessControlOriginators*. The following originator privilege types shall be considered for access control policy check.

Table 9.6.2.1-1: Types of Parameters in *accessControlOriginators*

Name	Description
domain	A SP domain or SP sub-domain
originatorIdentifier	CSE-Id or AE-ID which represent a originator identity
all	Any originators are allowed to access the resource within the *accessControlOriginators* constraints

9.6.2.2 *accessControlContexts*

The *accessControlContexts* is an optional parameter in an access-control-rule-tuple that contains a list, where each element of the list, when present, represents a context that is permitted to use this access control rule. Each request context is described by a set of parameters, where the types of the parameters can vary within the set. Table 9.6.2.2-1 describes the supported types of parameters in *accessControlContexts*.

The following originator *accessControlContexts* shall be considered for access control policy check by the CSE.

Table 9.6.2.2-1: Types of Parameters in *accessControlContexts*

Name	Description
accessControlTimeWindow	Represents a time window constraint which is compared against the time that Authorization Decision is made.
accessControlLocationRegion	Represents a location region constraint which is compared against the location of the Originator of the request
accessControlIpAddress	Represents an IP address constraint or IP address block constraint which is compared against the IP address of the Originator of the request

9.6.2.3 *accessControlOperations*

The *accessControlOperations* is a mandatory parameter in an access-control-rule-tuple that represents the set of operations that are authorized using this access control rule. Table 9.6.2.3-1 describes the supported set of operations that are authorized by *accessControlOperations*.

The following *accessControlOperations* shall be considered for access control policy check by the CSE..

Table 9.6.2.3-1: Types of parameters in *accessControlOperations*

Name	Description
RETRIEVE	Privilege to retrieve the content of an addressed resource
CREATE	Privilege to create a child resource
UPDATE	Privilege to update the content of an addressed resource
DELETE	Privilege to delete an addressed resource
DISCOVER	Privilege to discover the resource
NOTIFY	Privilege to receive a notification

9.6.3 Resource Type *CSEBase*

A <*CSEBase*> resource shall represent a CSE. The <*CSEBase*> resource shall be the root for all resources that are residing in the CSE.

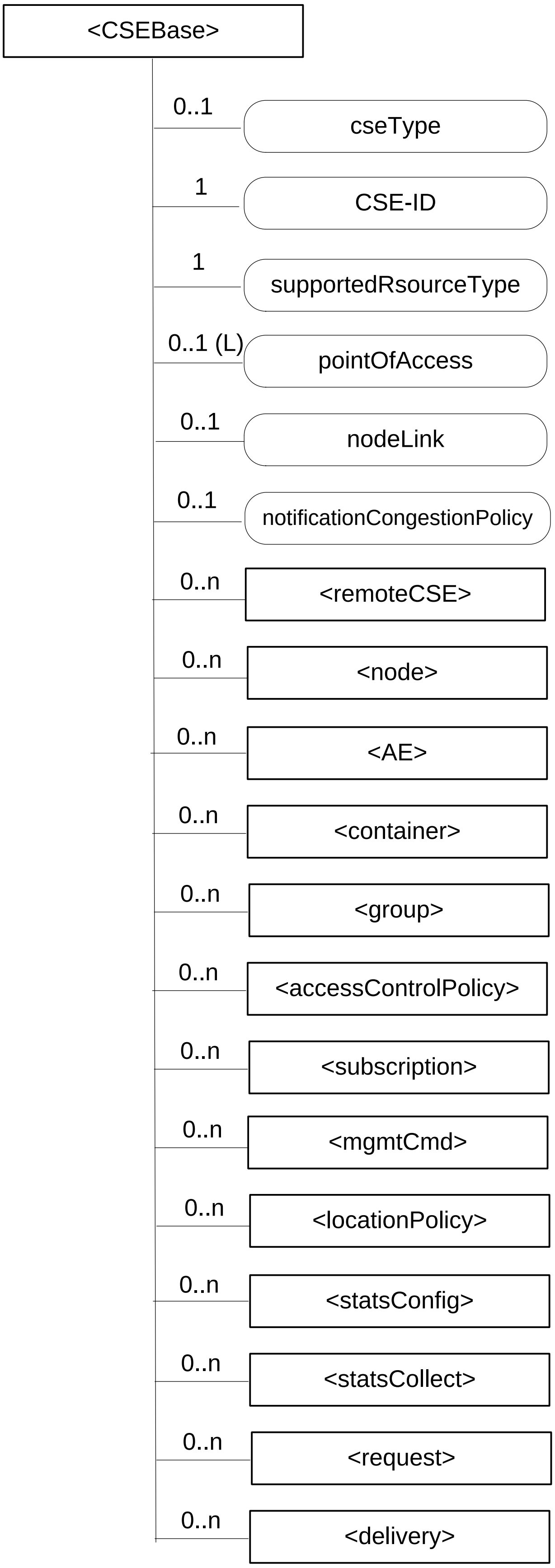


Figure 9.6.3-1: Structure of <*CSEBase*> resource

The <*CSEBase*> resource shall contain the child resources specified in table 9.6.3-1.

Table 9.6.3-1: Child resources of <*CSEBase*> resource

Child Resources of <*CSEBase*>	Child Resource Type	Multiplicity	Description
[variable]	*<remoteCSE>*	0..n	See clause 9.6.4
[variable]	*<node>*	0..n	See clause 9.6.18
[variable]	*<AE>*	0..n	See clause 9.6.5
[variable]	*<container>*	0..n	See clause 9.6.6
[variable]	*<group>*	0..n	See clause 9.6.13
[variable]	*<accessControlPolicy>*	0..n	See clause 9.6.2
[variable]	*<subscription>*	0..n	See clause 9.6.8
[variable]	*<mgmtCmd>*	0..n	See clause 9.6.16
[variable]	*<locationPolicy>*	0..n	See clause 9.6.10
[variable]	*<statsConfig>*	0..n	See clause 9.6.22
[variable]	*<statsCollect>*	0..n	See clause 9.6.24
[variable]	*<request>*	0..n	See clause 9.6.12
[variable]	*<delivery>*	0..n	See clause 9.6.11

The <*CSEBase*> resource shall contain the attributes specified in table 9.6.3-2.

Table 9.6.3-2: Attributes of <*CSEBase*> resource

Attributes of <*CSEBase*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
labels	0..1	RW	See clause 9.6.1 where this common attribute is described.
cseType	0..1	WO	Indicates the type of CSE represented by the created resource. • Mandatory for an IN-CSE, hence multiplicity (1). • Its presence is subject to SP configuration in case of an ASN-CSE or a MN-CSE.
CSE-ID	1	WO	The globally unique CSE identifier.
supportedResourceType	1	RO	List of the resource types which are supported in the CSE. This attribute contains subset of resource types listed in clause 9.2. For each supported *resourceType* this attribute indicates the supported optional attributes also.
pointOfAccess	0..1 (L)	RW	Represents the list of physical addresses to be used by remote CSEs to connect to this CSE (e.g. IP address, FQDN). This attribute is used to announce its value to remote CSEs.
nodeLink	0..1	RO	A reference (URI) of a <*node*> resource that stores the node specific information.
notificationCongestionPolicy	0..1	RO	This attribute applies to CSEs generating subscription notifications. It specifies the rule which is applied when the storage of notifications for each subscriber (an AE or CSE) reaches the maximum storage limit for notifications for that subscriber. E.g. Delete stored notifications of lower *notificationStoragePriority* to make space for new notifications of higher *notificationStoragePriority*, or delete stored notifications of older *creationTime* to make space for new notifications when all notifications are of the same *notificationStoragePriority*.

9.6.4 Resource Type *remoteCSE*

A <*remoteCSE*> resource shall represent a remote CSE that is registered to the Registrar CSE. <*remoteCSE*> resources shall be located directly under the <*CSEBase*> resource.

Similarly each registered CSE shall also be represented as a <*remoteCSE*> resource in the registering CSE's <*CSEBase*>.

For example, when CSE1 registers with CSE2, there will be two <*remoteCSE*> resources created: one in CSE1: <CSEBase1>/<remoteCSE2> and one in CSE2: <CSEBase2>/<remoteCSE1>.

Note that the creation of the two resources does not imply mutual registration. The <CSEBase1>/<remoteCSE2> does not mean CSE2 registered with CSE1 in the example above.

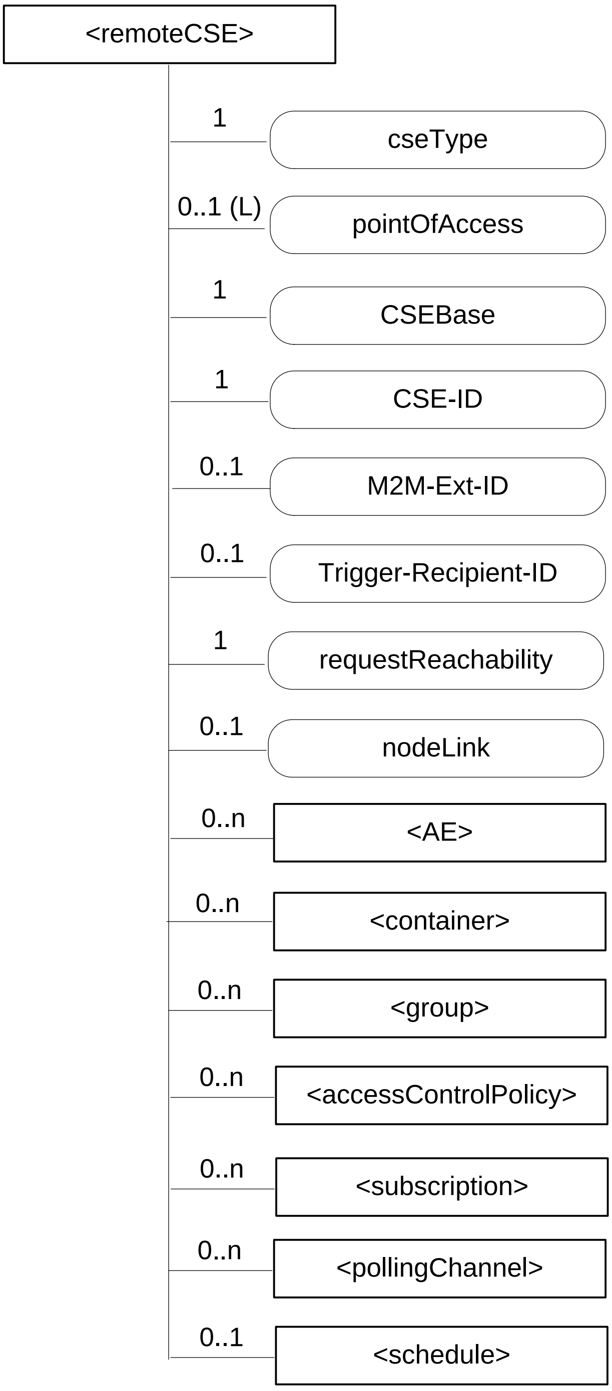


Figure 9.6.4-1: Structure of <*remoteCSE*> resource

The <*remoteCSE*> resource shall contain the child resources specified in table 9.6.4-1.

Table 9.6.4-1: Child resources of <*remoteCSE*> resource

Child Resources of <*remoteCSE*>	Child Resource Type	Multiplicity	Description	<*remoteCSEAnnc*> Child Resource Types
[variable]	<*AE*>	0..n	See clause 9.6.5	*AE* *<AEAnnc>*
[variable]	<*container*>	0..n	See clause 9.6.6	<*container*> <*containerAnnc*>
[variable]	<*group*>	0..n	See clause 9.6.13	<*group*> <*groupAnnc*>
[variable]	<*accessControlPolicy*>	0..n	See clause 9.6.2	<*accessControlPolicy*> <*accessControlPolicyAnnc*>
[variable]	<*subscription*>	0..n	See clause 9.6.8	<*subscription*>
[variable]	<*pollingChannel*>	0..n	See clause 9.6.21. If *requestReachability* is FALSE, the CSE that created this <*remoteCSE*> resource should create a <*pollingChannel*> resource and perform long polling.	<*pollingChannel*>
[variable]	<*schedule*>	0..1	This resource defines the reachability schedule information of the node. See clause 9.6.9 for <*schedule*>.	<*scheduleAnnc*>

The <remoteCSE> resource shall contain the attributes specified in table 9.6.4-2.

Table 9.6.4-2: Attributes of *<remoteCSE>* resource

Attributes of *<remoteCSE>*	Multiplicity	RW/ RO/ WO	Description	*<remoteCSEAnnc>* Attributes
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.	NA
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.	MA
parentID	1	RO	See clause 9.6.1 where this common attribute is described.	NA
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.	NA
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.	NA
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.	MA
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.	MA
labels	0..1	RW	See clause 9.6.1 where this common attribute is described.	MA
announceTo	1	RW	See clause 9.6.1 where this common attribute is described.	NA
announcedAttribute	1	RW	See clause 9.6.1 where this common attribute is described.	NA
cseType	0..1	WO	Indicates the type of CSE represented by the created resource. • Mandatory for an IN-CSE, hence multiplicity (1). • Its presence is subject to SP configuration in case of an ASN-CSE or a MN-CSE.	OA
pointOfAccess	0..1 (L)	RW	For request-reachable remote CSE it represents the list of physical addresses to be used to connect to it (e.g. IP address, FQDN). The attribute is absent if the remote CSE is not request-reachable.	OA
CSEBase	1	WO	The URI of the *CSEBase* of the original CSE represented by remote CSE.	OA
CSE-ID	1	WO	The globally unique CSE identifier.	OA
M2M-Ext-ID	0..1	RW	Supported when Registrar is IN-CSE. See clause 7.1.8 where this attribute is described. This attribute is used only for the case of dynamic association of M2M-Ext-ID and CSE-ID.	NA
Trigger-Recipient-ID	0..1	RW	Supported when Registrar is IN-CSE.. See clause 7.1.10 where this attribute is described. This attribute is used only for the case of dynamic association of M2M-Ext-ID and CSE-ID.	NA
requestReachability	1	RW	If the CSE that created this *<remoteCSE>* resource can receive a request from other AE/CSE(s), this attribute is set to "TRUE" otherwise "FALSE". NOTE: Even if this attribute is set to "FALSE", it does not mean it AE/CSE is always unreachable by all entities. E.g. the requesting AE/CSE is behind the same NAT, so it can communicate within the same NAT.	OA

Attributes of *<remoteCSE>*	Multiplicity	RW/ RO/ WO	Description	*<remoteCSEAnnc>* Attributes
nodeLink	0..1	RO	Reference URI of a <*node*> resource that stores node specific information.	OA

<*remoteCSE*> and the announced <*remoteCSE*> resources shall have different resourceType coding.

9.6.5 Resource Type *AE*

An <*AE*> resource represents information about an Application Entity known to a given CSE.

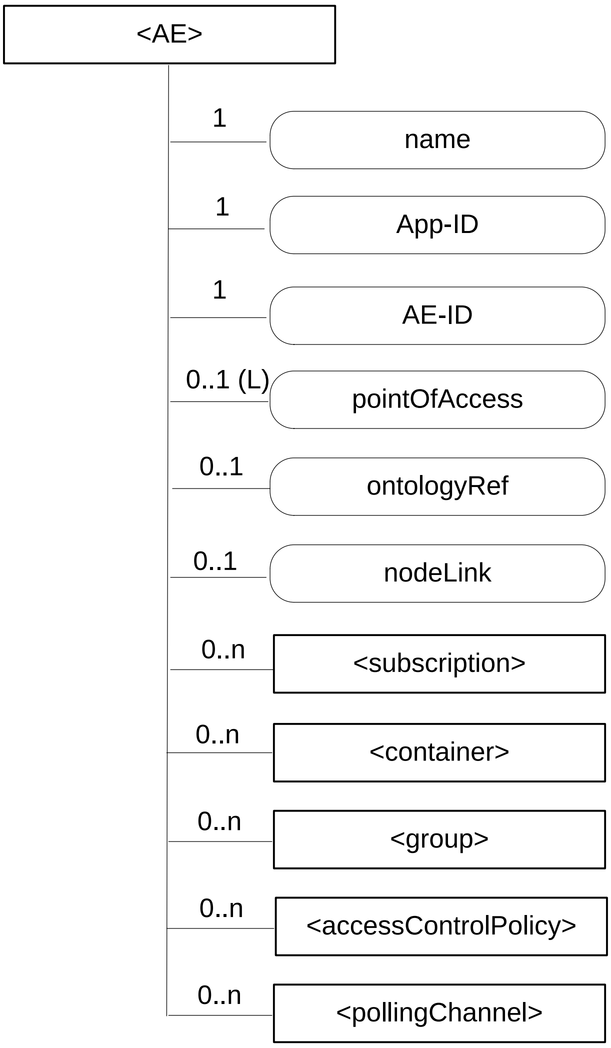


Figure 9.6.5-1: Structure of <*AE*> resource

The <*AE*> resource shall contain the child resources specified in table 9.6.5-1.

Table 9.6.5-1: Child resources of <*AE*> resource

Child Resources of <*AE*>	Child Resource Type	Multiplicity	Description	<*AEAnnc*> Child Resource Types
[variable]	<*subscription*>	0..n	See clause 9.6.8	<*subscription*>
[variable]	<*container*>	0..n	See clause 9.6.6	<*container*> <*containerAnnc*>
[variable]	<*group*>	0..n	See clause 9.6.13	<*group*> <*groupAnnc*>
[variable]	<*accessControlPolicy*>	0..n	See clause 9.6.2	<*accessControlPolicy*> <*accessControlPolicyAnnc*>
[variable]	<*pollingChannel*>	0..n	See clause 9.6.21. When the registrar CSE of this AE is request-unreachable, the AE should create this <*pollingChannel*> resource and perform long polling.	<*pollingChannel*>

The <*AE*> resource shall contain the attributes specified in table 9.6.5-2.

Table 9.6.5-2: Attributes of <*AE*> resource

Attributes of <*AE*>	Multiplicity	RW/ RO/ WO	Description	<*AEAnnc*> Attributes
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.	NA
parentID	1	RO	See clause 9.6.1 where this common attribute is described.	NA
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.	MA
accessControlPolicyIDs	1 (L)	RW	See clause 9.6.1 where this common attribute is described.	MA
creationTime	1	RW	See clause 9.6.1 where this common attribute is described.	NA
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.	NA
labels	0..1	RO	See clause 9.6.1 where this common attribute is described.	MA
announceTo	1	RW	See clause 9.6.1 where this common attribute is described.	NA
announcedAttribute	1	RW	See clause 9.6.1 where this common attribute is described.	NA
name	1	RO	The (usually human readable) name of the application, as declared by the application developer(e.g. "HeatingMonitoring")	OA
App-ID	1	RO	Application Identifier	OA
AE-ID	1	RO	The identifier of the Application Entity (see clause 7.1.2).	OA
pointOfAccess	0..1 (L)	RW	The list of addresses for communicating with the registered Application Entity over Mca reference point via the transport services provided by Underlying Network (e.g. IP address, FQDN, URI). This attribute shall be accessible only by the AE and the hosting CSE.	OA
ontologyRef	0..1	RW	A reference (URI) of the ontology used to represent the information that is managed and understood by the AE; to be passed to the AE.	OA
nodeLink	0..1	RO	A reference (URI) of a <*node*> resource that stores the node specific information.	OA

9.6.6 Resource Type *container*

The <*container*> resource represents a container for data instances. It is used to share information with other entities and potentially to track the data. A <*container*> resource has no associated content. It has only attributes and child resources.

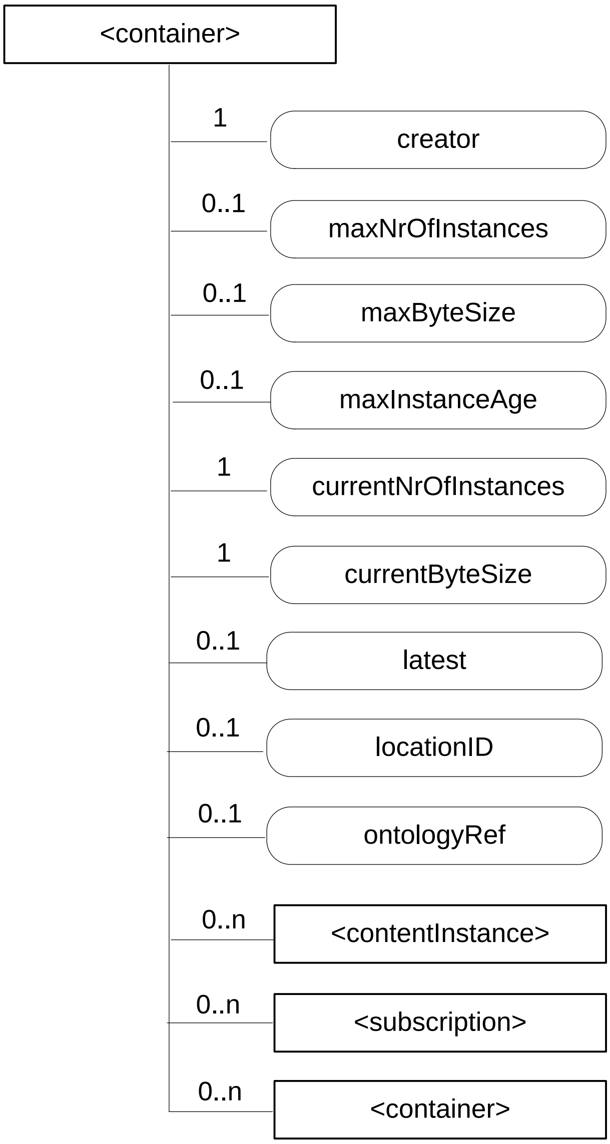


Figure 9.6.6-1: Structure of <*container*> resource

The <*container*> resource shall contain the child resources specified in table 9.6.6-1.

Table 9.6.6-1: Child resources of <*container*> resource

Child Resources of <*container*>	Child Resource Type	Multiplicity	Description	<*containerAnnc*> Child Resource Types
[variable]	<*contentInstance*>	0..n	See clause 9.6.7	<*contentInstance*>
[variable]	<*subscription*>	0..n	See clause 9.6.8	<*subscription*>
[variable]	<*container*>	0..n	See clause 9.6.6	<*container*> <*containerAnnc*>

The <*container*> resource shall contain the attributes specified in table 9.6.6-2.

Table 9.6.6-2: Attribute of <*container*> resource

Attributes of <*container*>	Multiplicity	RW/ RO/ WO	Description	<*containerAnnc*> Attributes
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.	NA
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.	MA
parentID	1	RO	See clause 9.6.1 where this common attribute is described.	NA
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.	MA
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described. If no *accessControlPolicyIDs* are provided at the time of creation, the *accessControlPolicyIDs* of the parent resource is linked to this attribute	MA
labels	0..1	RO	See clause 9.6.1 where this common attribute is described.	MA
creationTime	1	RW	See clause 9.6.1 where this common attribute is described.	NA
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.	NA
stateTag	1	RO	See clause 9.6.1 where this common attribute is described.	OA
announceTo	1	RW	See clause 9.6.1 where this common attribute is described.	NA
announcedAttribute	1	RW	See clause 9.6.1 where this common attribute is described.	NA
creator	1	RW	The AE-ID or CSE-ID of the entity which created the resource.	NA
maxNrOfInstances	0..1	RW	Maximum number of instances of <*contentInstance*> child resources.	OA
maxByteSize	0..1	RW	Maximum number of bytes that are allocated for a <*container*> resource for all instances in the <*container*> resource.	OA
maxInstanceAge	0..1	RW	Maximum age of the instances of <*contentInstance*> resources within the <*container*>. The value is expressed in seconds.	OA
currentNrOfInstances	1	RO	Current number of instances in a <*container*> resource. It is limited by the *maxNrOfInstances*.	OA
currentByteSize	1	RO	Current size in bytes of data stored in a <*container*> resource. It is limited by the *maxNrOfBytes*.	OA
latest	0..1	RO	Reference to latest <*contentInstance*> resource, when present.	OA
locationID	0..1	RW	URI of the resource where the attributes/policies that define how location information are obtained and managed. This attribute is defined only when the <*container*> resource is used for containing location information.	OA
ontologyRef	0..1	RW	A reference (URI) of the ontology used to represent the information that is stored in the instances of the container. NOTE: The access to this URI is out of scope of oneM2M.	OA

9.6.7 Resource Type *contentInstance*

The <*contentInstance*> resource represents a data instance in the <container> resource. The content of the *contentInstance* can be encrypted.

Unlike other resources, the <*contentInstance*> resource shall not be modified once created. An AE shall be able to delete a *contentInstance* resource explicitly or it may be deleted by the platform based on policies. If the platform has policies for *contentInstance* retention, these shall be represented by the attributes *maxByteSize*, *maxNrOfInstances* and/or *maxInstanceAge* attributes in the <*container*> resource. If multiple policies are in effect, the strictest policy shall apply.

The <*contentInstance*> resource inherits the same access control policies of the parent <*container*> resource, and does not have its own *accessControlPolicyIDs* attribute.

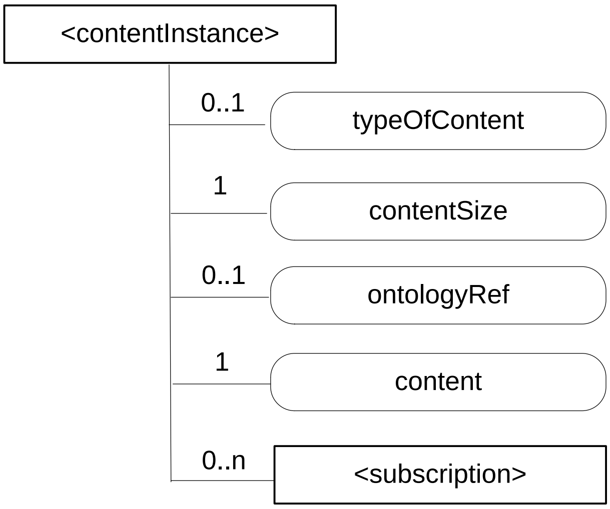


Figure 9.6.7-1: Structure of <*contentInstance*> resource

The <*contentInstance*> resource shall contain the child resources specified in table 9.6.7-1.

Table 9.6.7-1: Child resources of <*contentInstance*> resource

Child Resources of <*contentInstance*>	Child Resource Type	Multiplicity	Description
[variable]	*<subscription>*	0..n	See clause 9.6.8

The <contentInstance> resource shall contain the attributes specified in table 9.6.7-2.

Table 9.6.7-2: Attributes of *<contentInstance>* resource

Attributes of *<contentInstance>*	**Multiplicity**	**RW/ RO/ WO**	**Description**	***<contentInstanceAnnc>* Attributes**
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.	NA
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.	MA
parentID	1	RO	See clause 9.6.1 where this common attribute is described.	NA
labels	0..1	WO	See clause 9.6.1 where this common attribute is described.	MA
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.	NA
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.	NA
stateTag	1	RO	See clause 9.6.1 where this common attribute is described. The *stateTag* attribute of the parent resource should be incremented first and copied into this *stateTag* attribute when a new instance is added to the parent resource.	OA
announceTo	1	RW	See clause 9.6.1 where this common attribute is described.	NA
announcedAttribute	1	RW	See clause 9.6.1 where this common attribute is described.	NA
typeOfContent	0..1	WO	Type of the content included in the *content* attribute. This is media-type as defined in [i.2].	OA
contentSize	1	WO	Size in bytes of the *content* attribute.	OA
ontologyRef	0..1	WO	A reference (URI) of the ontology used to represent the information that is stored in the *contentInstances* resources of the *<container> resource*. If this attribute is not present, the *contentInstance* resource inherits the *ontologyRef* from the parent *<container>* resource if present NOTE: Access to this URI is out of scope of oneM2M.	OA
content	1	WO	Actual opaque content of a *contentInstance*. This may for example be an image taken by a security camera, or a temperature measurement taken by a temperature sensor.	OA

9.6.8 Resource Type *subscription*

The <*subscription*> resource contains subscription information for its subscribed-to resource. The subscribed-to resource is the resource that has the <*subscription*> resource as its child resource.

The <*subscription*> resource shall be represented as child resource of the subscribed-to resource. For example, <*container*> resource has <*subscription*> resource as a child resource (see clause 9.6.6). A <*subscription*> resource shall be deleted when the parent subscribed-to resource is deleted.

The <*subscription*> resource shall represent a subscription to a subscribed-to resource. An originator shall be able to create a resource of <*subscription*> resource type when the originator has RETRIEVE privilege to the subscribe-to resource. The originator of a <*subscription*> resource becomes the resource subscriber.

When a modification to the subscribed-to resource occurs, that modification is compared to the *notificationCriteria* attribute to determine whether a notification needs to be sent. If it matches, a Notify request shall be sent to *notificationURI(s)* in the <*subscription*> resource. When a <*subscription*> resource is deleted, a Notify request shall be sent to the *subscriberURI* if it is provided by the originator.

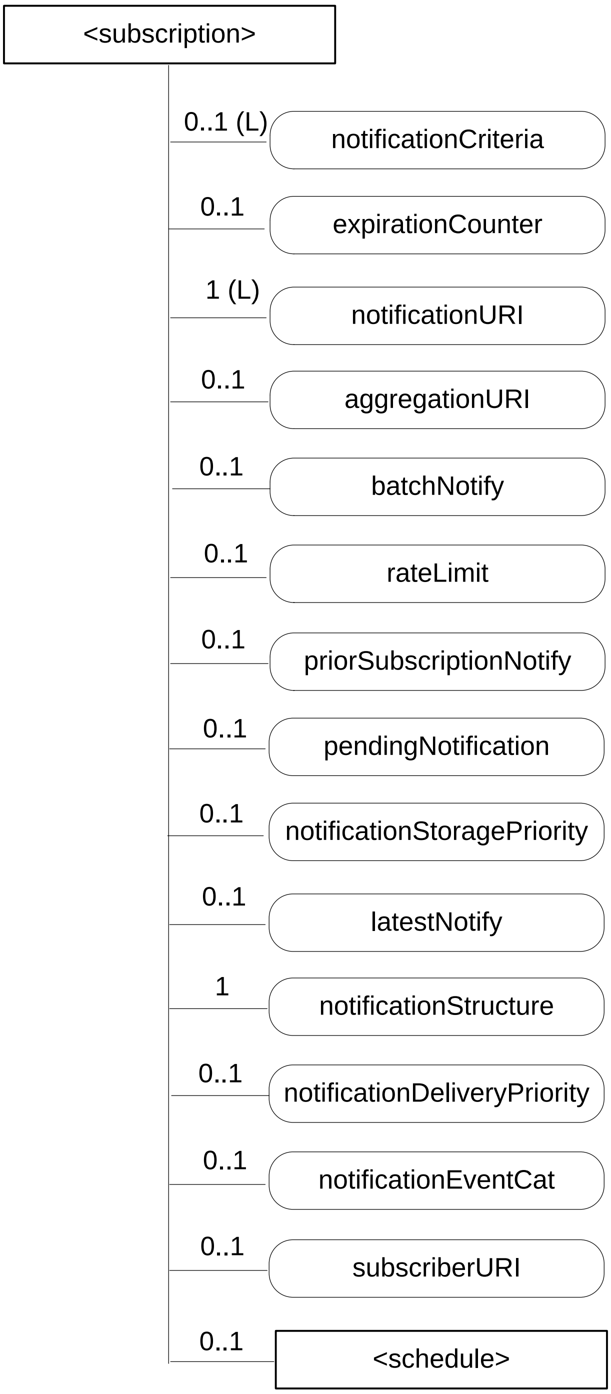


Figure 9.6.8-1: Structure of <*subscription*> resource

The <*subscription*> resource shall contain the child resources specified in table 9.6.8-1.

Table 9.6.8-1: Child resources of <*subscription*> resource

Child Resources of <*subscription*>	Child Resource Type	Multiplicity	Description
notificationSchedule	<*schedule*>	0..1	See clause 9.6.9

The <*subscription*> resource shall contain the attributes specified in table 9.6.8-2.

Table 9.6.8-2: Attributes of <*subscription*> resource

Attributes of <*subscription*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	WO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	l	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described. If no *accessControlPolicyIDs* is given at the time of creation, the *accesControlPolicies* of the parent resource is linked to this attribute.
notificationCriteria	0..1 (L)	RW	When notification events happen on the subscribed-to resource, the list of notification events that match the notification criteria shall be sent as a Notify request.
expirationCounter	0..1	RW	When the number of notifications becomes the same as this counter, the <*subscription*> resource shall be deleted.
notificationURI	1 (L)	RW	List of URI(s) where the resource subscriber will receive notifications. This list of URI(s) may not represent the resource subscriber entity.
aggregationURI	0..1	RW	URI to aggregate notifications from group members of a <*group*> resource.
batchNotify	0..1	RW	Indicates that notifications should be batched for delivery. When set, notification events are temporarily stored until either a specified number is ready to send or until a duration after the first notification event has expired.
rateLimit	0..1	RW	Indicates that notifications should be rate-limited. When set, notification events that exceed a specified number within a specified time are temporarily stored then sent when the number of events sent per specified time falls below the limit.
priorSubscriptionNotify	0..1	WO	Indicates that when this subscription is created, whether notification events prior to subscription should be sent, e.g. send prior "n" notifications, if available.
pendingNotification	0..1	RW	Indicates the notification action to be taken following a period of unreachability (according to the reachability schedule). When set, pending notification(s) during an unreachable period are processed according to *pendingNotification*. The following provides the possible values for *pedingNotification*: • “sendNone” • “sendLatest” • “sendAllPending” The default behavior if this attribute is not set is to send no notification (“sendNone”). When *sendLatest* is selected, the ***ec*** of the corresponding outgoing notification shall be set to *latest*.
notificationStoragePriority	0..1	RW	Indicates a priority for this subscription relative to other subscriptions belonging to this same subscriber for retention of notification events when storage is congested. The storage congestion policy which uses this attribute as input is specified in clause 9.6.3.
latestNotify	0..1	RW	Indicates if the subscriber wants only the latest notification. If no notifications are buffered at the hosting CSE or transit CSE, then all notifications will be received. If the notifications are buffered, and if the value of this attribute is set to true, then older notifications shall be discarded. The attribute is mutual exclusive with *batchNotify*.

Attributes of *<subscription>*	Multiplicity	RW/ RO/ WO	Description
notificationStructure	1	RW	Type of information that shall be contained in notifications. E.g. modified attribute only of a subscribed-to resource, a whole subscribed-to resource, and/or URI of a corresponding <*subscription*> resource.
notificationDeliveryPriority	0..1	RW	Indicates a delivery priority for the notification. That is, this attribute defines the how to handle sending of notifications when notifications need to be sent.
notificationEventCat	0..1	RW	Defines the Event Categories for the notification message triggered by the <*subscription*> resource.
subscriberURI	0..1	WO	URI that the <*subscription*> created entity can get notification from the <*subscription*> hosting CSE. <*subscription*> deletion shall be notified if this URI is provided.

The *notificationCriteria* attribute shall be configured a priori in the <*subscription*> resource. If an event matches the *notificationCriteria* conditions, then a notification shall be delivered.

Table 9.6.8-3 describes the *notificationCriteria* conditions.

Table 9.6.8-3: *notificationCriteria* conditions

Condition tag	Multiplicity	Matching condition
createdBefore	0..1	The *creationTime* attribute of the resource is chronologically before the specified value.
createdAfter	0..1	The *creationTime* attribute of the resource is chronologically after the specified value.
modifiedSince	0..1	The *lastModifiedTime* attribute of the resource is chronologically after the specified value.
unmodifiedSince	0..1	The *lastModifiedTime* attribute of the resource is chronologically before the specified value.
stateTagSmaller	0..1	The *stateTag* attribute of the resource is smaller than the specified value.
stateTagBigger	0..1	The *stateTag* attribute of the resource is bigger than the specified value.
expireBefore	0..1	The *expirationTime* attribute of the resource is chronologically before the specified value.
expireAfter	0..1	The *expirationTime* attribute of the resource is chronologically after the specified value.
labels	0..n	The *labels* attributes of the resource matches the specified value.
resourceType	0..n	The *resourceType* attribute of the child resource of the subscribed-to resource is the same as the specified value. It allows notification of child resource creation and deletion.
sizeAbove	0..1	The *contentSize* attribute of the <*contentInstance*> resource is equal to or greater than the specified value.
sizeBelow	0..1	The *contentSize* attribute of the <*contentInstance*> resource is smaller than the specified value.
contentType	0..n	The *typeOfContent* attribute of the <*contentInstance*> resource matches the specified value
resourceStatus	0..n	When the subscribed-to resource is changed by the operations or expiration, the resource status is the same as the specified value. Possible values are: child created, updated, child deleted, deleted.
operationMonitor	0..n	The operations accessing the subscribed-to resource matches with the specified value. It allows monitoring which operation is attempted to the subscribed-to resource regardless of whether the operation is performed. This feature is useful when to find malicious AEs. Possible string arguments are: create, retrieve, update, delete.
attribute	0..n	This is an attribute of resource types (clause 9.6). Therefore, a real tag name is variable depends on its usage. E.g., *creator* of container resource type can be used as a filter criteria tag as "creator=Sam".

9.6.9 Resource Type *schedule*

The <*schedule*> resource contains scheduling information.

The <*schedule*> resource shall represent the scheduling information in the context of its parent resource. An originator shall have the same access control privileges to the <*schedule*> resource as it has to its parent resource.

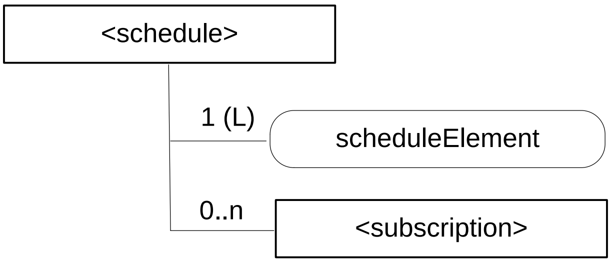


Figure 9.6.9-1: Structure of <*schedule*> resource

The <schedule> resource shall contain the child resource specified in table 9.6.9-1.

Table 9.6.9-1: Child resources of <*schedule*> resource

Child Resources of <*schedule*>	Child Resource Type	Multiplicity	Description	<*scheduleAnnc*> Child Resource Types
[variable]	<*subscription*>	0..n	See clause 9.6.8	None

The <*schedule*> resource shall contain the attributes specified in table 9.6.9-2.

Table 9.6.9-2: Attributes of <*schedule*> resource

Attributes of <*schedule*>	Multiplicity	RW/ RO/ WO	Description	<*scheduleAnnc*> Attributes
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.	NA
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.	MA
parentID	1	RO	See clause 9.6.1 where this common attribute is described.	NA
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.	MA
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.	NA
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.	NA
labels	0..1	RO	See clause 9.6.1 where this common attribute is described.	MA
announceTo	1	RW	See clause 9.6.1 where this common attribute is described.	NA
announcedAttribute	1	RW	See clause 9.6.1 where this common attribute is described.	NA
scheduleElement	1 (L)	RW	Expresses time periods defined by second, minute, hour day of month, month, and year. Supports repeating periods, and wildcards expressed as a list.	OA

9.6.10 Resource Type *locationPolicy*

The <*locationPolicy*> resource represents the method for obtaining and managing geographical location information of an M2M Node.

The actual location information shall be stored in a <*contentInstance*> resource which is a child resource of the <*container*> resource. The <*container*> resource includes the *locationId* attribute which holds the URI of this <*locationPolicy*> resource. A CSE can obtain location information based on the attributes defined under <*locationPolicy*> resource, and store the location information in the target <*container*> resource.

Based on the *locationSource* attribute, the method for obtaining location information of an M2M Node can be differentiated. The methods for obtaining location information shall be as follows:

- **Network-based method:** where the CSE on behalf of the AE obtains the target M2M Node's location information from an Underlying Network.

- **Device-based method:** where the ASN is equipped with any location capable modules or technologies (e.g. GPS) and is able to position itself.

- **Sharing-based method:** where the ADN has no GPS nor an Underlying Network connectivity. Its location information can be retrieved from either the associated ASN or a MN.

NOTE: Geographical location information could include more than longitude and latitude.

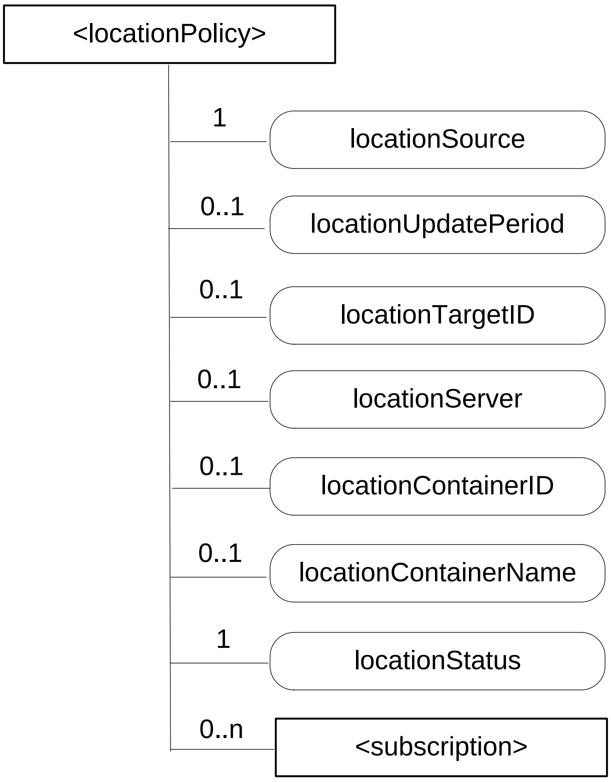


Figure 9.6.10-1: Structure of *<locationPolicy>* resource

The *<locationPolicy>* resource shall contain the child resources specified in table 9.6.10-1.

Table 9.6.10-1: Child resources of *<locationPolicy>* resource

Child Resources of *<locationPolicy>*	Child Resource Type	Multiplicity	Description	*<locationPolicyAnnc>* Child Resource Types
[variable]	*<subscription>*	0..n	See clause 9.6.8	None

The <*locationPolicy*> resource shall contain the attributes specified in table 9.6.10-2.

Table 9.6.10-2: Attributes of <*locationPolicy*> resource

Attributes of <*locationPolicy*>	Multiplicity	RW/ RO/ WO	Description	<*locationPolicyAnnc*> Attributes
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.	NA
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.	MA
parentID	1	RO	See clause 9.6.1 where this common attribute is described.	NA
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.	MA
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.	MA
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.	NA
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.	NA
labels	0..1	RW	See clause 9.6.1 where this common attribute is described.	MA
announceTo	1	RW	See clause 9.6.1 where this common attribute is described.	NA
announcedAttribute	1	RW	See clause 9.6.1 where this common attribute is described.	NA
locationSource	1	RW	Indicates the source of location information • Network Based • Device Based • Sharing Based	OA
locationUpdatePeriod	0..1	RW	Indicates the period for updating location information. If the value is marked '0' or not defined, location information is updated only when a retrieval request is triggered.	OA
locationTargetId	0..1	RW	The identifier to be used for retrieving the location information of a remote Node and this attribute is only used in the case that location information is provided by a location server.	OA
locationServer	0..1	RW	Indicates the identity of the location server. This attribute is only used in that case location information is provided by a location server.	OA
locationContainerID	0..1	RO	A URI of the <*container*> resource where the actual location information of a M2M Node is stored.	OA
locationContainerName	0..1	RW	A Name of the <*container*> resource where the actual location information of a M2M Node is stored. If it is not assigned, the hosting CSE automatically assigns a name of the resource. Note: The created <*container*> resource related to this policy shall be stored only in the hosting CSE.	OA
locationStatus	1	RO	Contains the information on the current status of the location request. (e.g., location server fault)	OA

9.6.11 Resource Type *delivery*

When a CSE is requested to initiate an operation (CRUDN) targeting resources on another CSE, then it needs to do scheduling and execution of delivery of data from the source CSE to the target CSE in line with the provisioned policies. It shall be in one of the following ways:

- Using delivery aggregation (**da** information set to ON), or
- Forwarding the original request as a separate request on the Mcc reference point without changes.

In order to be able to initiate and manage the execution of data delivery in a resource-based manner, resource type <*delivery*> is defined. This resource type shall be used for forwarding requests from one CSE to another CSE when the **da** parameter in the request is set to ON. If the **da** parameter is set to OFF, the original request shall be forwarded without change to the next CSE, i.e. without the use of <*delivery*> resource. If the **da** parameter is not present, the latter method shall be used.

Operations to Retrieve, Update or Delete a <*delivery*> resource shall allow authorized entities to inquire the status of a delivery, change delivery attributes or cancel a delivery.

As defined in clause 10.2.4, <*delivery*> resource can only be created by a CSE. A request for the creation of a <*delivery*> resource can only be issued to a registrar CSE. <*delivery*> resource is deleted on successful delivery of the data in the *aggregatedRequest* attribute to the next hop CSE.

The parent of a <*delivery*> resource is the <*CSEBase*> resource of the CSE that accepted the request for the creation of the <*delivery*> resource.

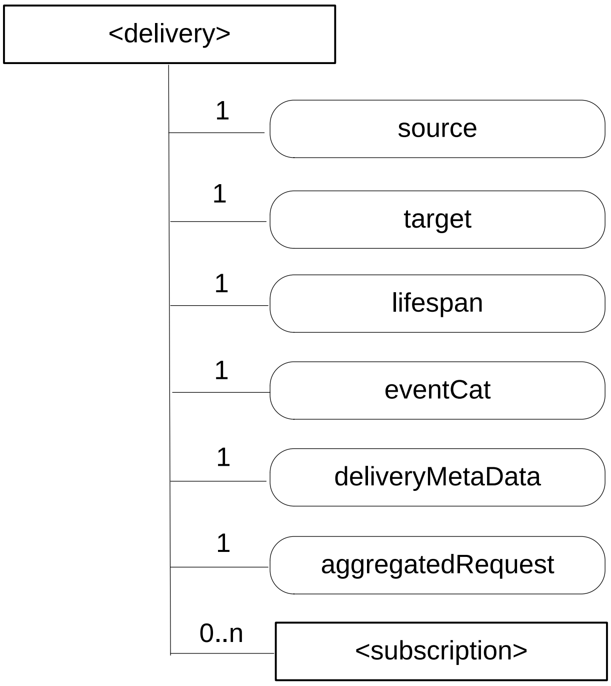


Figure 9.6.11-1: Structure of <*delivery*> resource

The <*delivery*> resource shall contain the child resource specified in table 9.6.11-1.

Table 9.6.11-1: Child resources of <*delivery*> resource

Child Resources of <*delivery*>	Child Resource Type	Multiplicity	Description
[variable]	*<subscription>*	0..n	See clause 9.6.8

The <*delivery*> resource shall contain the attributes specified in table 9.6.11-2.

Table 9.6.11-2: Attributes of <*delivery*> resource

Attributes of <*delivery*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described.
stateTag	1	RO	See clause 9.6.1 where this common attribute is described.
source	1	WO	The CSE-ID of the CSE that initiated the delivery process represented by this <delivery> resource.
target	1	WO	CSE-ID that defines the hosting CSE for delivering the data contained in the *aggregatedRequest* attribute
lifespan	1	RW	Defines the time limit when the delivery of the information in the *aggregatedRequest* attribute needs to complete. If the *lifespan* expires before successful delivery, no further attempts to deliver the information in the *aggregatedRequest* attribute need to be executed. If the delivery fails, a feedback may be expected by the source CSE depending on options reflected in the *deliveryMetaData* attribute. The *lifespan* attribute of a <*delivery*> resource shall be set consistent with the **rqet** parameters of the set of original requests contained in the *aggregatedRequest* attribute, i.e. *lifespan* shall not extend beyond the earliest expiring **rqet** parameter in the set of the original requests contained in the *aggregatedRequest* attribute.
eventCat	1	RW	Defines the category of the event that triggered the delivery request represented by this <*delivery*> resource.
deliveryMetaData	1	RW	Contains meta information on the delivery process represented by this <*delivery*> resource, such as delivery status, delivery options, tracing information, etc
aggregatedRequest	1	WO	Attribute containing the request(s) to be delivered to the hosting CSE. This represents one or more original requests that were targeting the same hosting CSE.

9.6.12 Resource Type *request*

The use of <*request*> resource type is optional depending on the configuration.

Creation of a <*request*> resource can only be done on a registrar CSE implicitly when a registered AE or a registered CSE issues a request to the registrar CSE targeting any other resource type or requesting a notification. Creation of a <*request*> resource instance is only permitted by the registrar CSE as a result of a request from an originator which contains the **rt** parameter in the request message and where **rt** parameter is set to *'nonBlockingReqeustSynch'* or *'nonBlockingRequestAsynch'*.

When a CSE is requested to initiate an operation for which the result should be available to the originator by reference (**rt** information of the request set to *'nonBlockingReqeustSynch'* or *'nonBlockingRequestAsynch'*), the registrar CSE which received the request directly from the originator shall provide a reference of the created <*request*> resource back to the originator so that the originator can access attributes of the <*request*> at a later time - for instance in order to retrieve the result of an operation that was taking a longer time. If the registrar CSE uses resources of type <*request*> to keep such context information, the reference that shall be given back to the originator as part of the acknowledgment that is the address of the <*request*> resource. The originator (or any other authorized entity depending on access control) can access the request status and the requested operation result through it.

The <*request*> resource may be deleted by the CSE that is hosting it when the expiration time of the <*request*> resource is reached. So after the expiration time of a <*request*> resource is reached it cannot be assumed that that particular <*request*> resource is still accessible. Depending on implementation of the CSE that is hosting it, a <*request*> resource may also get deleted earlier than the expiration time, when the result of the requested operation (if any result was requested at all) has been sent back to the originator.

For the purpose of providing a standardized structure for expressing and accessing the context of a previously issued request, the resource type <*request*> is defined. The parent resource of a <*request*> resource shall be the <*CSEBase*> resource of the hosting CSE.

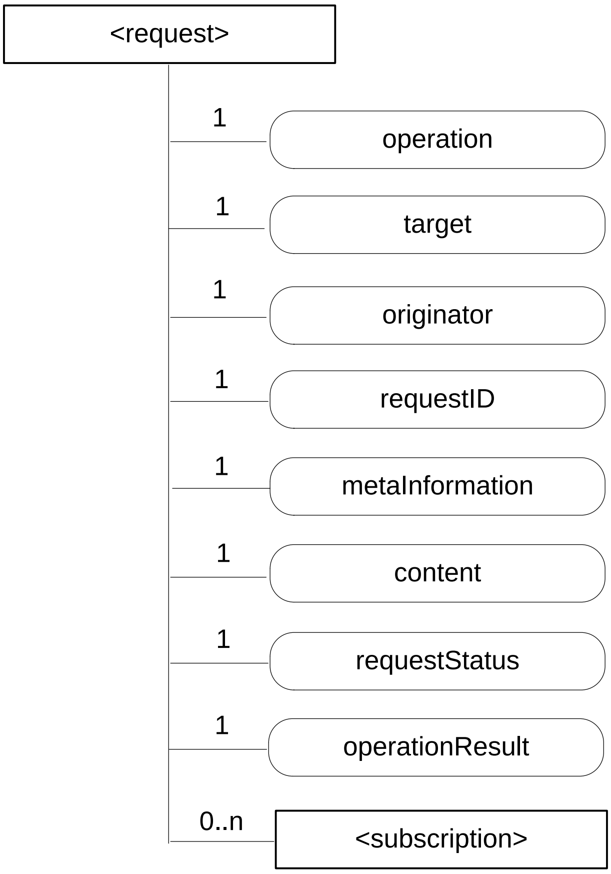


Figure 9.6.12-1: Structure of <*request*> resource

The <*request*> resource shall contain the child resources specified in table 9.6.12-1.

Table 9.6.12-1: Child resources of <*request*> resource

Child Resources of <*request*>	Child Resource Type	Multiplicity	Description
[variable]	<*subscription*>	0..n	See clause 9.6.8

The <*request*> resource shall contain the attributes specified in table 9.6.12-2.

Table 9.6.12-2: Attributes of *<request>* resource

Attributes of *<request>*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described. The value of the expirationTime is chosen by the CSE dependent on the **rqet**, **rset**, **rp** and **oet** parameters associated with the original request.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described.
stateTag	1	RO	See clause 9.6.1 where this common attribute is described.
operation	1	RO	It contains the value of the parameter **op** in the original request message.
target	1	RO	It contains the value of the parameter **to** in the original request message.
originator	1	RO	It contains the value of the parameter **fr** in the original request message.
requestID	1	RO	It contains the value of the parameter **ri** in the original request message.
metaInformation	1	RO	Meta information about the request. The content of this attribute is equivalent to information in any other optional parameters described in clause 8.1.
content	1	RO	Contains the content that is carried in the **cn** parameter of the original request message.
requestStatus	1	RO	Contains information on the current status of the Request, e.g. "accepted and pending".
operationResult	1	RO	Contains the result of the originally requested operation in line with the **rc** parameter associated with the original request.

All operations on <*request*> resources except for the CREATE operations - which can only be triggered implicitly by a request for which a <*request*> resource shall capture the context - are controlled by the access control policy.

9.6.13 Resource Type *group*

The <*group*> resource represents a group of resources of the same or mixed types. The <*group*> resource can be used to do bulk manipulations on the resources represented by the *membersList* attribute. The <*group*> resource contains an attribute that represents the members of the group and a virtual resource (the <*fanOutPoint*>*)* that allows operations to be applied to the resources represented by those members.

When used as one of the permission holders in an <*accessControlPolicy*> resource, the group can be used to grant a collection of AEs (represented by <*AE*> resources) or CSEs (represented by <*remoteCSE*> resources) permissions for accessing (e.g. creating child resources, retrieving, etc.) a resource.

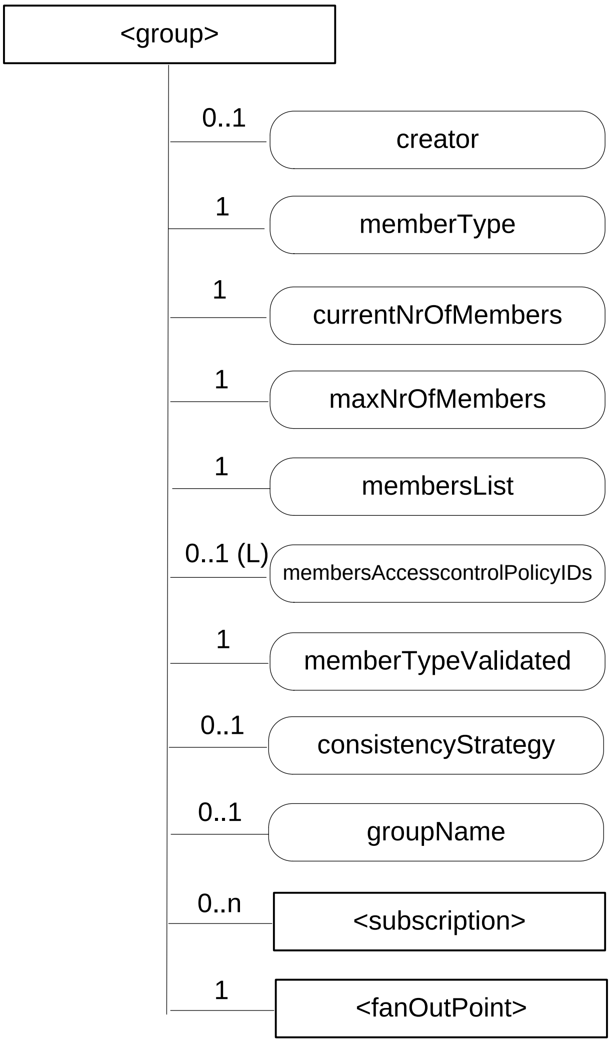


Figure 9.6.13-1: Structure of <*group*> resource

The <*group*> resource shall contain the child resources specified in table 9.6.13-1.

Table 9.6.13-1: Child resources of <group> resource

Child Resources of <*group*>	Child Resource Type	Multiplicity	Description	<*groupAnnc*> Child Resource Types
[variable]	*<subscription>*	0..n	See clause 9.6.8	none
fanOut	*<fanOutPoint>*	1	See clause 9.6.14	none

The *<group>* resource shall contain the attributes specified in table 9.6.13-2

Table 9.6.13-2: Attributes of <*group*> resource

Attributes of <*group*>	Multiplicity	RW/ RO/ WO	Description	<*groupAnnc*> Attributes
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.	NA
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.	MA
parentID	1	RO	See clause 9.6.1 where this common attribute is described.	NA
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.	MA
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.	MA
labels	0..1	RO	See clause 9.6.1 where this common attribute is described.	MA
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.	NA
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.	NA
announceTo	1	RW	See clause 9.6.1 where this common attribute is described.	NA
announcedAttribute	1	RW	See clause 9.6.1 where this common attribute is described.	NA
creator	0..1	RW	The AE-ID or CSE-ID of the entity which created the resource.	NA
memberType	1	WO	It is the resource type of the members resources of the group if all members resources (including the members resources in any sub-groups) are of the same type. Otherwise, it's a type of 'mixed'.	OA
currentNrOfMembers	1	RO	Current number of members in a group. It is limited by the *maxNrOfMembers*.	OA
maxNrOfMembers	1	RW	Maximum number of members in the <*group*>.	OA
membersList	1	RW	List of zero or more member URIs referred to in the remaining of this specification as *memberID*. Each URI (*memberID*) should refer to a members resource or a (sub-) <*group*> resource of the <*group*>.	OA
membersAccessControlPolicyIDs	0..1 (L)	RW	List of URIs of the <*accessControlPolicy*> resources defining who is allowed to access the <*fanOutPoint*> resource.	OA
memberTypeValidated	1	RO	Denotes if *memberType* of all members resources of the group has been validated.	OA
consistencyStrategy	0..1	WO	This attribute determines how to deal with the <*group*> resource if the *memberType* validation fails. Which means delete the inconsistent member if the attribute is ABANDON_MEMBER; delete the group if the attribute is ABANDON_GROUP; set the *memberType* to "mixed" if the attribute is SET_MIXED.	OA
groupName	0..1	RW	Human readable name of the <*group*>.	OA

9.6.14 Resource Type *fanOutPoint*

The <*fanOutPoint*> resource is a virtual resource because it does not have a representation. It is the child resource of a <*group*> resource. Whenever a request is sent to the <*fanOutPoint*> resource, the request is fanned out to each of the members of the <*group*> resource indicated by the *membersList* attribute of the <*group*> resource. The responses (to the request) from each member are then aggregated and returned to the originator.

The <*fanOutPoint*> resource does not have a resource representation by itself and consequently it does not have an *accessControlPolicyIDs* attribute. The <*accessControlPolicy*> resource used for access control policy validation is indicated by the *membersAccessControlPolicyIDs* attribute in the parent <*group*> resource.

9.6.15 Resource Type *mgmtObj*

The <*mgmtObj*> resource contains management data which represents individual M2M management functions. It represents a general structure to map to external management technology e.g. OMA DM [i.5], BBF TR-069 [i.4] and LWM2M [i.6] data models. Each instance of <*mgmtObj*> resource shall be mapped to single external management technology.

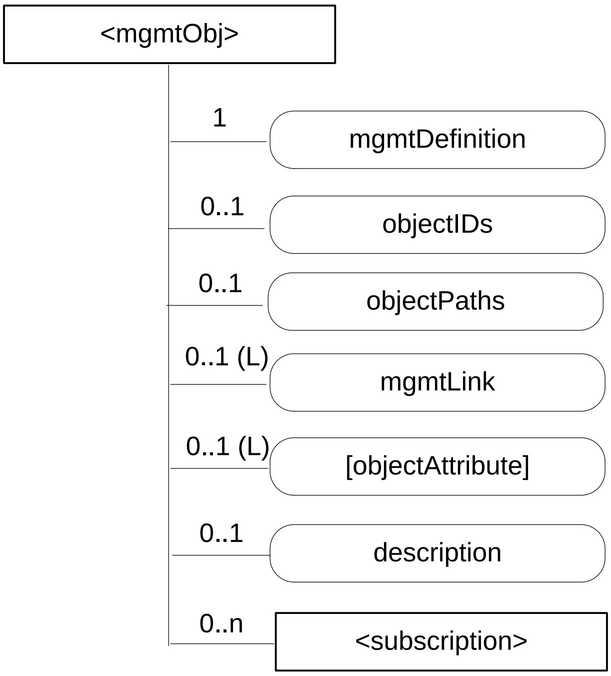


Figure 9.6.15-1: Structure of <*mgmtObj*> resource

The <*mgmtObj*> resource shall contain the child resource specified in table 9.6.15-1.

Table 9.6.15-1: Child resources of <*mgmtObj*> resource

Child Resources of <*mgmtObj*>	Child Resource Type	Multiplicity	Description
[variable]	*<subscription>*	0..n	See clause 9.6.8

The <*mgmtObj*> resource shall contain the attributes specified in table 9.6.15-2.

Table 9.6.15-2: Attributes of <*mgmtObj*> resource

Attributes of <*mgmtObj*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	WO	See clause 9.6.1 where this common attribute is described
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described

Attributes of *<mgmtObj>*	Multiplicity	RW/ RO/ WO	Description
accessControlPolicyIDs	1 (L)	RW	See clause 9.6.1 where this common attribute is described
creationTime	1	RO	See clause 9.6.1 where this common attribute is described
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described
labels	0..1	RO	See clause 9.6.1 where this common attribute is described
mgmtDefinition	1	WO	Specifies the type of <*mgmtObj*> resource e.g. software, firmware, memory. The list of the value of the attribute can be seen in Annex D.
objectIDs	0..1	WO	Contains the list URNs that uniquely identify the external management technology data models used for this <*mgmtObj*> resource as well as the managed function and version it represents. This attribute shall be provided during the creation of the <*mgmtObj*> resource and shall not be modifiable afterwards. If the <*mgmtObj*> resource is mapped to multiple external management technology data models, this attribute shall list all URNs for each mapped external management technology data model. This is mandatory for the <*mgmtObj*>, for which the data model is not specified by oneM2M but mapped from external management technology data model specified in other device management protocols.
objectPaths	0..1	WO	Contains the list of local paths of the external management object instances on the managed entity which is represented by the <*mgmtObj*> resource in the hosting CSE. This attribute shall be provided during the creation of the <*mgmtObj*>, so that the hosting CSE can correlate the created <*mgmtObj*> with the external management object instance on the managed entity for further management operations. It shall not be modifiable after creation. The format of this attribute shall be a local external management object path in the form as specified by existing management protocols. (e.g. "./anyPath/Fw1" in OMA DM [i.5], "Device.USBHosts.Host.3." in BBF TR-069 [i.4]). The combination of the *objectPath* and the *objectID* attribute, allows to address the external technology data models.
mgmtLink	0..1 (L)	RW	This attribute contains reference to a list of other <*mgmtObj*> resources in case a hierarchy of <*mgmtObj*> is needed
[objectAttribute]	0..1 (L)	RW	Each *[objectAttribute]* is mapped from a leaf node of a hierarchical structured management object (including oneM2M data model and the existing management data models) based on the mapping rules below the table.
description	0..1	RW	Text format description of <*mgmtObj*>.

When mapping resources from management technologies to a corresponding <*mgmtObj*> resource, the following rules shall apply:

- The root resource of external management objects maps to the <*mgmtObj*> resource
- For the child resource of the root resource of specific technology:
  - **Rule1:** If the child external management object cannot have another child external management object, the external management object maps to the *[objectAttribute]* attribute of the <*mgmtObj*> resource with the same resource name.
  - **Rule2:** If the child external management object can have another child external management object, the external management object maps to a new <*mgmtObj*> resource. The URI of the new <*mgmtObj*> resource is stored as an *mgmtLink* attribute of the <*mgmtObj*> resource which is mapped from the parent external management object.

9.6.16 Resource Type *mgmtCmd*

The <*mgmtCmd*> resource represents a method to execute management procedures or to model commands and remote procedure calls (RPC) required by existing management protocols (e.g. BBF TR-069 [i.4]), and enables AEs to request management procedures to be executed on a remote entity. It also enables cancellation of cancellable and initiated but unfinished management procedures or commands.

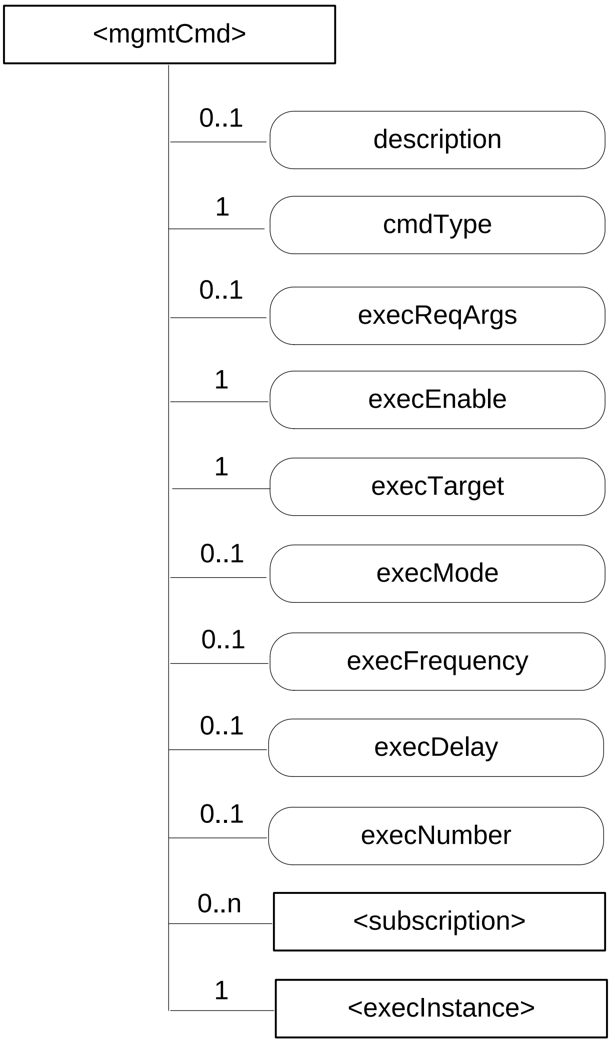


Figure 9.6.16-1: Structure of <*mgmtCmd*> resource

Each <*mgmtCmd*> corresponds to a specific type of management command, as defined by its attribute *cmdType*. For multiple requests of the same management command, <*mgmtCmd*> may use the child-resource (i.e. <*execInstance*>) to contain all execution instances. The execution of the management procedure represented by <*mgmtCmd*> shall be triggered using the UPDATE method to its attribute *execEnable*.

The <*mgmtCmd*> resource shall contain the child resources specified in table 9.6.16-1.

Table 9.6.16-1: Child resources of <*mgmtCmd*> resource

Child Resources of <*mgmtCmd*>	Child Resource Type	Multiplicity	Description
[variable]	<*subscription*>	0..n	See clause 9.6.8
[variable]	<*execInstance*>	1	See clause 9.6.17

The <*mgmtCmd*> resource shall contain the attributes specified in table 9.6.16-2.

Table 9.6.16-2: Attributes of <*mgmtCmd*> resource

Attributes of <*mgmtCmd*>	**Multiplicity**	**RW/ RO/ WO**	**Description**
resourceType	1	WO	See clause 9.6.1 where this common attribute is described
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described
labels	0..1	RW	See clause 9.6.1 where this common attribute is described
creationTime	1	RO	See clause 9.6.1 where this common attribute is described
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described
description	0..1	RW	The text-format description of this resource.
cmdType	1	WO	The type to identify the management operation (e.g. download).
execReqArgs	0..1	RW	Structured attribute (e.g. abstract type) to contain any command-specific arguments of the request.
execEnable	1	RO	The attribute can be blank without any value or it can contain a URI that can be used to trigger execution of <*mgmtCmd*> using UPDATE method.
execTarget	1	RW	M2M-Node-ID of the target on which this <*mgmtCmd*> will be executed.
execMode	0..1	RW	The mode used to specify how the command will be executed (e.g., Immediate Once, Immediate and Repeatedly, Random Once, Random and Repeatedly). May be used together with *execFrequency*, *execDelay* and *execNumber* to provide the scheduling information.
execFrequency	0..1	RW	The minimum interval between two executions, to be used in conjunction with *execMode*. Modes involving random execution can be used to add random values between individual executions.
execDelay	0..1	RW	The minimum delay before the instance should be executed. Modes involving random execution can be used to increase this delay randomly.
execNumber	0..1	RW	The number of times the instance should be executed, to be used when *execMode* indicates a repetition pattern.

[9.6.16.a] Editor's Note: The types of management operations (e.g. cmdType) to be used by mgmtCmd are FFS.

[9.6.16.b] Editor’s Note: It is FFS for how to use execTarget for group management

9.6.17 Resource Type *execInstance*

The <*execInstance*> resource represents a successful instance of <*mgmtCmd*> execution request, which had been triggered by a M2M network application using the UPDATE method to the attribute *execEnable* of <*mgmtCmd*> resource.

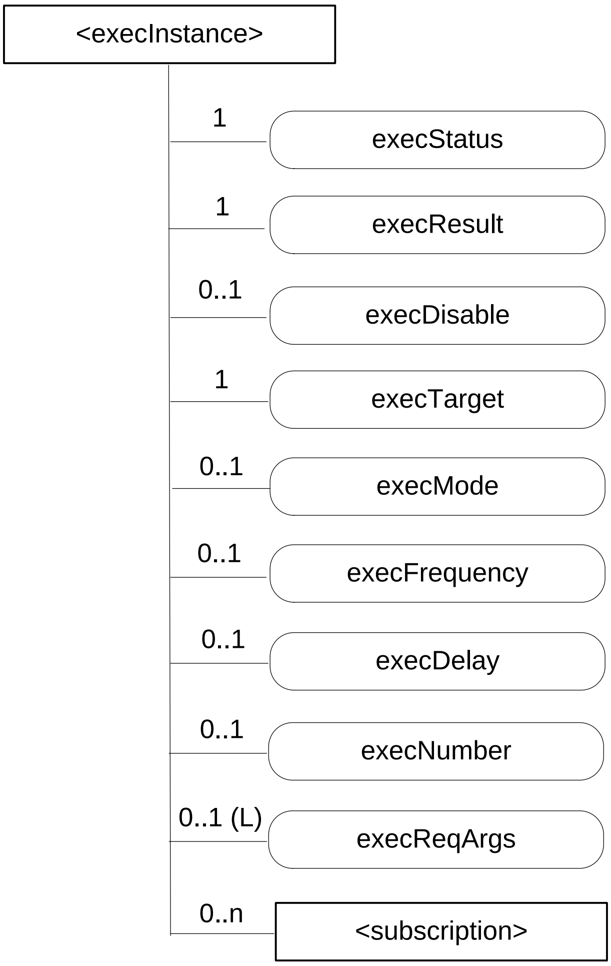


Figure 9.6.17-1: Structure of <*execInstance*> resource

The <*execInstance*> resource shall contain the child resources specified in table 9.6.17-1.

Table 9.6.17-1: Child resources of <*execInstance*> resource

Child Resources of <*execInstance*>	Child Resource Type	Multiplicity	Description
[variable]	<*subscription*>	0..n	See clause 9.6.8

The <*execInstance*> resource shall contain the attributes specified in table 9.6.17-2.

Table 9.6.17-2: Attributes of <*execInstance*> resource

Attributes of <*execInstance*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RO	See clause 9.6.1 where this common attribute is described
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	1 (L)	RW	See clause 9.6.1 where this common attribute is described
creationTime	1	RO	See clause 9.6.1 where this common attribute is described
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described
labels	0..1	RO	See clause 9.6.1 where this common attribute is described
execStatus	1	RO	The status of <*execInstance*>. It can be Initiated, Started, Finished, Cancelled, or Deleted.
execResult	1	RO	The execution result of <*execInstance*>.
execDisable	0..1	RW	The attribute is used to cancel <*execInstance*> using UPDATE method.
execTarget	1	RO	M2M-Node-ID of the target.
execMode	0..1	RO	Modes used to specify how the command will be executed (e.g. Immediate Once, Immediate and Repeatedly, Random Once, Random and Repeatedly). May be used together with *execFrequency*, *execDelay* and *execNumber* to provide the scheduling information.
execFrequency	0..1	RO	The minimum interval between two executions, to be used in conjunction with *execMode*. Modes involving random execution can be used to add random values between individual executions.
execDelay	0..1	RO	The minimum delay before the instance should be executed. Modes involving random execution can be used to increase this delay randomly.
execNumber	0..1	RO	The number of times the instance should be executed, to be used when *execMode* indicates a repetition pattern.
execReqArgs	0..1 (L)	RO	Structured attribute (e.g. abstract type) to contain any command-specific arguments (as a list) used to trigger this <*execInstance*>.

9.6.18 Resource Type *node*

The <*node*> resource represents specific information that provides properties of an M2M Node that can be utilized by other oneM2M operations. The <*node*> resource has <*mgmtObj*> as its child resources. The <*mgmtObj*> resources represent the Node's context information (e.g. memory and battery), network topology, device information, device capability etc. The <*mgmtObj*> resources are also resources to perform management of the Nodes.

This node specific information stored in this resource type such as memory and battery can be obtained either by the existing device management technologies (OMA DM [i.5], BBF TR-069 [i.4]) or any other way (e.g. JNI [i.21]). Since <*mgmtObj*> resource type represents the management functions including both ways, the types of sub-resources are <*mgmtObj*> resource type.

For the case when the <*node*> resource belongs to an ADN, please see the description of *nodeLink* attribute in the <*AE*> resource (clause 9.6.5).

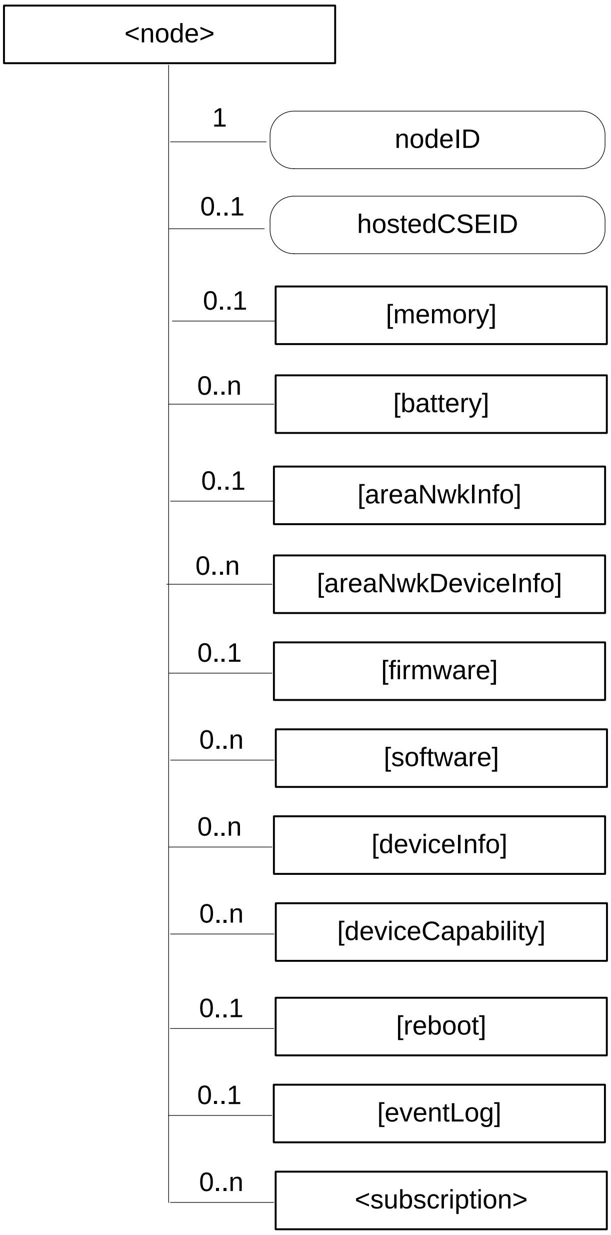


Figure 9.6.18-1: Structure of <*node*> resource

The *<node>* resource shall contain the child resources specified in table 9.6.18-1.

Table 9.6.18-1: Child resources of *<node>* resource

Child Resources of *<node>*	Child Resource Type	Multiplicity	Description
[variable]	*<mgmtObj> with mgmtDefinition of value "memory"*	0..1	This resource provides the memory (typically RAM) information of the node. (e.g. the amount of total volatile memory), See clause D.4.
[variable]	*<mgmtObj> with mgmtDefinition of value "battery"*	0..n	The resource provides the power information of the node. (e.g. remaining battery charge). See clause D.7.
[variable]	*<mgmtObj> with mgmtDefinition of value "areaNwkInfo"*	0..1	This resource describes the list of Nodes attached behind the MN node and its physical or underlying relation among the nodes in the M2M Area Network. This attribute is defined in case the Node is MN. See clause D.5.
[variable]	*<mgmtObj> with mgmtDefinition of value "areaNwkDeviceInfo"*	0..n	This resource describes the information about the Node in the M2M Area Network. See clause D.6.
[variable]	*<mgmtObj> with mgmtDefinition of value "firmware"*	0..1	This resource describes the information about the firmware of the Node include name, version etc, See clause D.2.
[variable]	*<mgmtObj> with mgmtDefinition of value "software"*	0..n	This resource describes the information about the software of the Node. See clause D.3.
[variable]	*<mgmtObj> with mgmtDefinition of value "deviceInfo"*	0..n	The resource contains information about the identity, manufacturer and model number of the device. See clause D.8.
[variable]	*<mgmtObj> with mgmtDefinition of value "deviceCapability"*	0..n	The resource contains information about the capability supported by the Node. See clause D.9.
[variable]	*<mgmtObj> with mgmtDefinition of value "reboot"*	0..1	The resource is the place to reboot or reset the Node. See clause D.10.
[variable]	*<mgmtObj> with mgmtDefinition of value "eventLog"*	0..1	The resource contains the information about the log of events of the Node. See clause D.11.
[variable]	*<subscription>*	0..n	See clause 9.6.8

The *<node>* resource shall contain the attributes specified in table 9.6.18-2.

Table 9.6.18-2: Attributes of *<node>* resource

Attributes of *<node>*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RW	See clause 9.6.1 where this common attribute is described.
nodeID	1	RW	The ID of the Node which is specified in the resource type. This attribute is identical to *M2M-Node-ID*.
hostedCSEID	0..1	RW	The ID of the *CSEBase* that is hosted on this node if type of node is ASN/MN/IN.

9.6.19 Resource Type *m2mServiceSubscription*

The <*m2mServiceSubscription*> resource represents an M2M Service Subscription. It is used to represent all data pertaining to the M2M Service Subscription, i.e., the technical part of the contract between an M2M Application Service Provider and an M2M Service Provider.

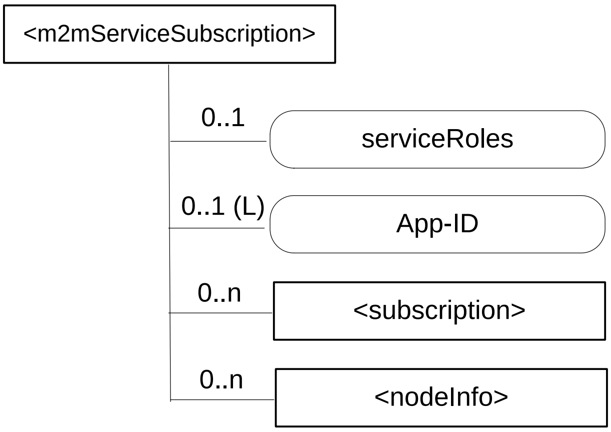


Figure 9.6.19-1: Structure of *<m2mServiceSubscription>* resource

The <*m2mServiceSubscription*> resource shall contain the child resources specified in table 9.6.19-1.

Table 9.6.19-1: Child resources of *<m2mServiceSubscription>* resource

Child Resources of *<m2mServiceSubscription>*	Child Resource Type	Multiplicity	Description
[variable]	*<subscription>*	0..n	See clause 9.6.8
nodeInfo	*<nodeInfo>*	0..n	See clause 9.6.20

The <*m2mServiceSubscription*> resource shall contain the attributes specified in table 9.6.19-2.

Table 9.6.19-2: Attributes of *<m2mServiceSubscription>* resource

Attributes of *<m2mServiceSubscription>*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described. If no *accessControlPolicyIDs* is given at the time of creation, the *accessControlPolicyIDs* of the parent resource is linked to this attribute
creationTime	1	RW	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
serviceRoles	0..1	RW	This attribute contains a list of Service Role IDs (S-RoleIDs) that are subscribed to in this service subscription. If the multiplicity of this attribute is 0, the role does not apply.
App-ID	0..1 (L)	RW	The value of this attribute shall be set to a list of application identifiers as defined in clause 7.1.3 pertaining to applications of this M2M service subscription. It may contain a wildcard, meaning that any App-ID is allowed.

9.6.20 Resource Type *nodeInfo*

The <*nodeInfo*> resource represents M2M Node information that is needed as part of the M2M Service Subscription resource. It shall contain information about the M2M Node as well as application identifiers of the Applications running on that Node.

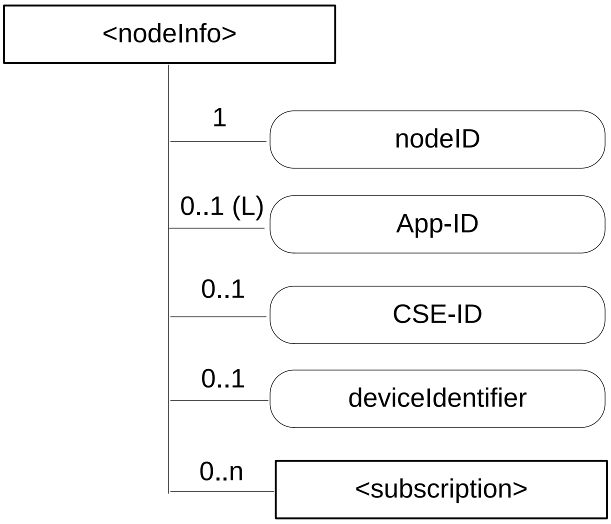


Figure 9.6.20-1: Structure of <*nodeInfo*> resource

The <*nodeInfo*> resource shall contain the child resource specified in table 9.6.20-1.

Table 9.6.20-1: Child resources of <*nodeInfo*> resource

Child Resources of <*nodeInfo*>	Child Resource Type	Multiplicity	Description
[variable]	*<subscription>*	0..n	See clause 9.6.8

The <*nodeInfo*> resource shall contain the attributes specified in table 9.6.20-2.

Table 9.6.20-2: Attributes of <*nodeInfo*> resource

Attributes of <*nodeInfo*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described. If no *accessControlPolicyIDs* is given at the time of creation, the *accessControlPolicyIDs* of the parent resource is linked to this attribute.
creationTime	1	RW	See clause 9.6.1 where this common attribute is described.
labels	0..1	RW	See clause 9.6.1 where this common attribute is described
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
nodeID	1	RW	Node identifier indicating where the M2M Service Subscription applies. A wildcard may be used to allow all the Nodes within a M2M Service Provider.
App-ID	0..1 (L)	RW	See clause 7.1.3 for definition of *App-Id*. *List* of Application IDs pertaining to the applications running on this node.
CSE-ID	0..1	RW	CSE-ID pertaining to this node (for nodes that have a CSE). The CSE-ID becomes know once the pre CSE registration procedure is triggered.
deviceIdentifier	0..1	RW	A list of device identifiers. A *deviceIdentifier* identifes a device using a Universally Unique IDentifier (UUID). The UUID specifies a valid, hex digit character string as defined in [RFC4122]. The format of the URN is urn:uuid:########-####-####-############ • **OPS URN:** Identify a device using the format <OUI> "-" <ProductClass> "-" <SerialNumber> as defined in Section 3.4.4 of TR-069 [i-4]. The format of the URN is urn:dev:ops:<OUI> "-" <ProductClass> "-" <SerialNumber>. • **OS URN:** Identify a device using the format <OUI> "-"<SerialNumber> as defined in Section 3.4.4 of TR-069 [i.4]. The format of the URN is urn:dev:os:<OUI> "-"<SerialNumber>. • **IMEI URN:** Identify a device using an International Mobile Equipment Identifiers of 3GPP-TS_23.003 [i.26]. The IMEI URN specifies a valid, 15 digit IMEI. The format of the URN is urn:imei:############### • **ESN URN:** Identify a device using an Electronic Serial Number. The ESN specifies a valid, 8 digit ESN. The format of the URN is urn:esn:######## • **MEID URN:** Identify a device using a Mobile Equipment Identifier. The MEID URN specifies a valid, 14 digit MEID. The format of the URN is urn:meid:##############

9.6.21 Resource Type *pollingChannel*

The <*pollingChannel*> resource represent a channel that can be used for a request-unreachable entity (i.e. an AE or a CSE which is behind NAT so it cannot receive a request from other Nodes). The request-unreachable entity creates a <*pollingChannel*> resource on a request-reachable CSE, and then polls any type of request(s) for itself from the <*pollingChannel*> hosting CSE. For example, an AE can retrieve notifications by long polling on the channel when it cannot receive notifications asynchronously from a subscription hosting CSE.

This <*pollingChannel*> resource shall have *longPollingURI* virtual attribute which is an input/output end point of the channel. After an AE or CSE creates this resource, it performs long polling on *longPollingURI* of this resource with RETRIEVE operation. The response to the long polling request is pending until there are any requests received on the channel.

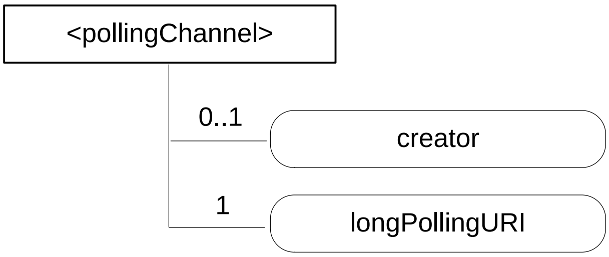


Figure 9.5.21-1: Structure of <*pollingChannel*> resource

The <*pollingChannel*> resource shall contain the attributes specified in table 9.6.21-1.

Table 9.6.21-1: Attributes of <*pollingChannel*> resource

Attributes of <*pollingChannel*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
labels	0..1	RW	See clause 9.6.1 where this common attribute is described.
creator	0..1	WO	The *AE-ID* of the entity which created the resource.
longPollingURI	1	WO	A virtual attribute that represents a polling URI. This URI is used for long polling. Accessing this attribute is intended to store request(s) (clause 10.2.13.6) and retrieve the request(s) (clause 10.2.13.7).

9.6.22 Resource Type *statsConfig*

The <*statsConfig*> resource is used to store configuration of statistics for AEs. The <*statsConfig*> resource may be established by the IN-CSEs or by AEs in IN-CSE. The <*statsConfig*> resource shall be located directly under <*CSEBase*>. <*eventConfig*> sub-resource shall be used to define events that trigger statistics collection. Below are some examples of events that can be generated:

- Collection based on a certain operation: collects any RETRIEVE operations on the data created by the collecting entity.

- Collection based on storage size: collects the size of storage when a <*container*> resource created by the collecting entity exceeds a quota.

- Combined configuration: collects all RETRIEVE operations on the data created by the collecting entity during a period of time.

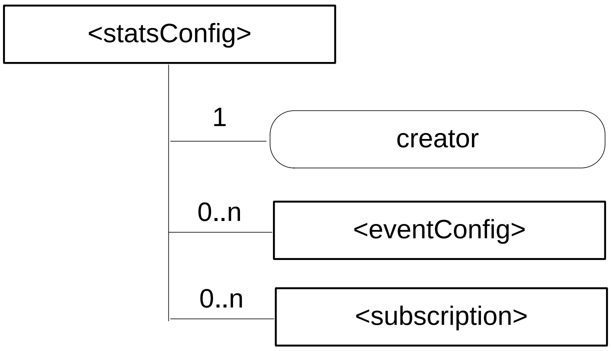


Figure 9.6.22-1: Structure of <*statsConfig*> resource

The <*statsConfig*> resource shall contain the child resources specified in table 9.6.22-1.

Table 9.6.22-1: Child resources of <*statsConfig*> resource

Child Resources of <*statsConfig*>	Child Resource Type	Multiplicity	Description
[variable]	<*eventConfig*>	0..n	See clause 9.6.23. This resource configures an event for statistics collection.
[variable]	<*subscription*>	0..n	See clause 9.6.8 where the type of this resource is described.

The <*statsConfig*> resource shall contain the attributes specified in table 9.6.22-2.

Table 9.6.22-2: Attributes of <*statsConfig*> resource

Attributes of <*statsConfig*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	1 (L)	RW	See clause 9.6.1 where this common attribute is described
creationTime	1	RO	See clause 9.6.1 where this common attribute is described
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described
labels	0..1	RW	See clause 9.6.1 where this common attribute is described
creator	1	RW	The *AE-ID* or *CSE-ID* of the entity which created the resource.

9.6.23 Resource Type *eventConfig*

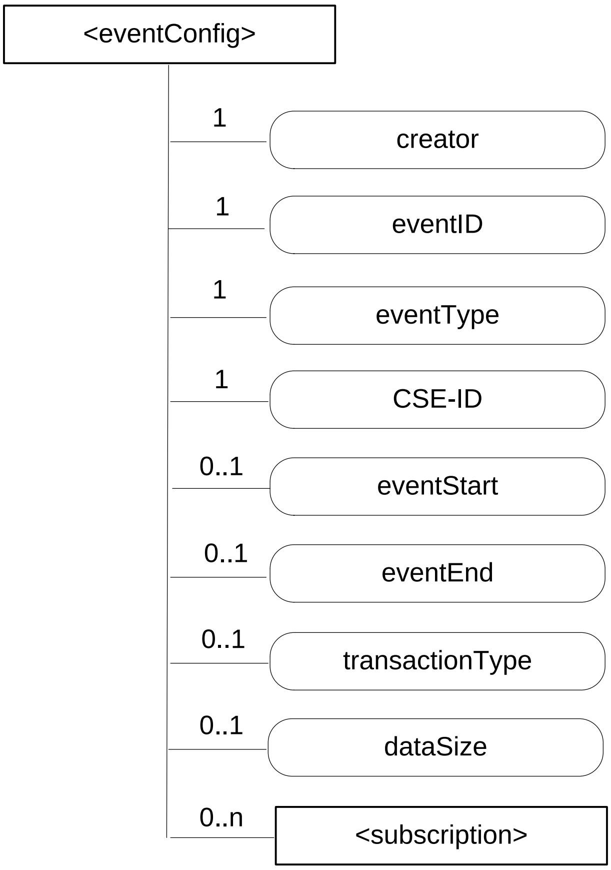


Figure 9.6.23-1: Structure of <*eventConfig*> resource

The <*eventConfig*> resource shall contain the child resource specified in table 9.6.23-1.

Table 9.6.23-1: Child resources of <*eventConfig*> resource

Child Resources of <*eventConfig*>	Child Resource Type	Multiplicity	Description
[variable]	<*subscription*>	0..n	See clause 9.6.8 where this type of resource is described.

The <*eventConfig*> resource shall contain the attributes specified in table 9.6.23-2.

Table 9.6.23-2: Attributes of <*eventConfig*> resource

Attributes of <*eventConfig*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	1 (L)	RW	See clause 9.6.1 where this common attribute is described
creationTime	1	RO	See clause 9.6.1 where this common attribute is described
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described
labels	0..1	RW	See clause 9.6.1 where this common attribute is described
creator	1	RW	The *AE-ID* or *CSE-ID* of the entity which created the resource.
eventID	1	RO	This attribute uniquely identifies the event to be collected for statistics for AEs.
eventType	1	RW	This attribute indicates the type of the event, such as timer based, data operation, storage based, etc.
eventStart	0..1	RW	This attribute indicates the start time of the event.
eventEnd	0..1	RW	This attribute indicates the end time of the event
transactionType	0..1	RW	This attribute defines the type of the operation to be collected by statistics, such as CREATE, RETRIEVE.
dataSize	0..1	RW	This attribute defines the data size if an event is triggered when the stored data exceeds a certain size.

9.6.24 Resource Type *statsCollect*

The <*statsCollect*> resource shall be used to collect information for AEs using the <*eventConfig*> resource as the triggers for the IN-CSE. The IN-CSE may setup multiple triggers. Each trigger may be activated or de-activated independently of others. The <*statsCollect*> resource shall be located directly under <CSEBase> of IN-CSE.

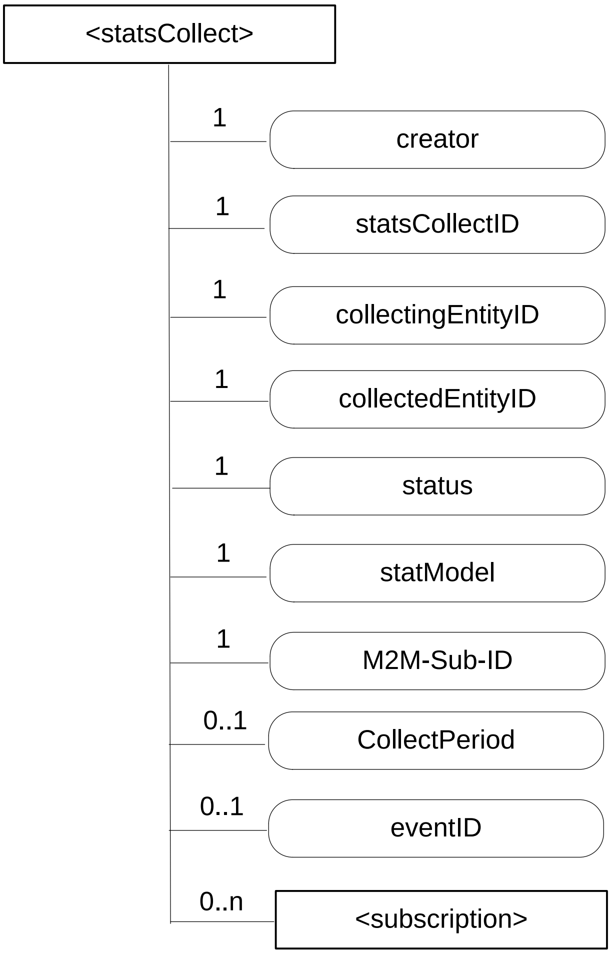


Figure 9.6.24-1: Structure of <*statsCollect*> resource

The <*statsCollect*> resource shall contain the child resource specified in table 9.6.24-1.

Table 9.6.24-1: Child resources of <*statsCollect*> resource

Child Resources of <*statsCollect*>	Child Resource Type	Multiplicity	Description
[variable]	<*subscription*>	0..n	See clause 9.6.8 where the type of this resource is described.

The <*statsCollect*> resource shall contain the attributes specified in table 9.6.24-2.

Table 9.6.24-2: Attributes of <*statsCollect*> resource

Attributes of <*statsCollect*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	1 (L)	RW	See clause 9.6.1 where this common attribute is described
creationTime	1	RO	See clause 9.6.1 where this common attribute is described
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described
labels	0..1	RW	See clause 9.6.1 where this common attribute is described
creator	1	RW	The *AE-ID* or *CSE-ID* of the entity which created the resource.
statsCollectID	1	RO	This is the unique ID to identify a specific statistics collection scenario. It is created by the IN-CSE when the <*statsCollect*> resource is first created.
collectingEntityID	1	WO	This is the unique ID of the entity that requests the collection of statistics. For example, it can be an *AE-ID* or *CSE-ID*.
collectedEntityID	1	WO	This is the unique ID of the entity whose operations at IN-CSE will be collected. For example, it can be an *AE-ID* or *CSE-ID*. If no specific value is provided for this attribute, the IN-CSE interprets it as "any entity".
status	1	RW	This attribute indicates whether the rule is "active" or "inactive".
statModel	1	RW	This attribute indicates the collection model, such as "Subscriber based", "event based", etc.
M2M-Sub-ID	1	WO	This attribute indicates the service subscription Identifier (*M2M-Sub-ID*) being recorded.
collectPeriod	0..1	RW	This attribute defines the duration to collect service statistics information.
eventID	0..1	RW	This attribute refers to the <*eventConfig*> resource that defines the events that can be collected by the IN-CSE. It is mandatory if the *statmodel* attribute is set to "event based".

9.6.25 Resource Announcement

A resource can be announced to one or more remote CSEs to inform the remote CSEs of the existence of the original resource. An announced resource can have a limited set of attributes and a limited set of child resources from the original resource. The announced resource includes a link to the original resource hosted by the original resource-hosting CSE.

In case that the original resource is deleted, all announced resources for the original resource shall be deleted. If the announced resource is not deleted promptly (e.g. the announced resource is not reachable), the announced resource can be deleted later either by the original resource hosting CSE or by the expiration of the announced resource itself. The original resource shall store the list of links for the announced resources for those purposes.

Synchronization between the attributes announced by the original resource and the announced resource is the responsibility of the original resource hosting CSE. The access control policy for the announced resource shall synchronize with the one from the original resource. In case that the attribute *accessControlPolicyIDs* is not present in the original resource it is the responsibility of the original resource hosting CSE to choose the appropriate value depending on the policy for the original resource (e.g. take the parent *accessControlPolicyIDs* value).

The original resource shall have at least *announceTo* attribute present if the resource itself has been announced. If any of the Optional Announced **(OA)** attributes are also announced, then *announcedAttribute* attribute shall also be present. An AE or other CSE can request the original resource hosting CSE for announcing the original resource to the list of CSE-IDs or the URI(s) listed in the *announceTo* attribute in the announcing request. An Update to the *announceTo* attribute will trigger new resource announcement(s) or the de-announcement(s) of the announced resource. After a successful announcement procedure the attribute *announceTo* contains only the list of URI(s) of the announced resources.

In order to announce the attribute marked as **OA** (see clause 9.5.1.1), they shall be included in the *announcedAttribute* attribute at the original resource. The attributes included in the *announcedAttribute* attribute are announced to the announced resource. On successful announcement of the resource, such attributes shall be created at the announced resource; otherwise they shall not be present in the announced resource. Update to the *announcedAttribute* attribute in the original resource will trigger new attribute announcement or the de-announcement of the announced attribute(s). The announced attributes shall have the same value as the original resource, and synchronization between the value of the announced attributes at the original resource and the announced resource is the responsibility of the original resource hosting CSE.

An announced resource may have child resources. The child resources shall be of the same type of the original child resources or their associated Announce type.

Child resources of the original resource can be announced independently as needed. In this case, the child resources at the announced resource shall be of the child resource Announced Type. When a child resource at the announced resource is created locally at the remote CSE, the child resource shall be of normal child resource type.

9.6.25.1 Common Attributes for Announced Resources

Table 9.6.25-1 lists the common attributes for the announced resources.

Table 9.6.25-1: Common Attributes for Announced Resources

Common Attributes	Mandatory /Optional	Description
resourceType	Mandatory	Resource Type. As specified in clause 9.2, a suffix of "**Annc**" to the name of the original resource type shall be used to indicate the name for the associated announced resource type.
resourceID	Mandatory	Resource Identifier See clause 9.6.1 for further information on this attribute.
parentID	Mandatory	Identifies the parent resource.
accessControlPolicyIDs	Mandatory	The list of identifiers (either an ID or a URI) of an <*accessControlPolicy*> resource announced by the original resource See clause 9.6.1 for further information on this attribute. If this attribute was not present in the original resource, the original resource shall include this attribute by providing the *accessControlPolicyIDs* from the original resource's parent resource or from the local policy according at the original resource.
creationTime	Mandatory	See clause 9.6.1 for information on this attribute.
expirationTime	Mandatory	See clause 9.6.1 for information on this attribute. This attribute is limited by the value received from the original resource but it can be overridden by the policies of the remote CSE hosting the announced resource.
lastModifiedTime	Mandatory	See clause 9.6.1 for information on this attribute.
labels	Conditionally Mandatory	Tokens used as keys for discovering resources as announced by the original resource. See clause 9.6.1 for further information on this attribute. The attribute is conditionally mandatory, which means that the attribute shall exist in the announced resource if it is present in the original resource.
link	Mandatory	Provides the URI to the original resource.

This is a draft oneM2M document and should not be relied upon; the final version, if any, will be made available by oneM2M Partners Type 1.

10 Information Flows

10.1 Basic Procedures

As a pre-condition to the execution of the following procedures, M2M operational security procedures as specified in clauses 11.3.1 through 11.3.3 shall have been performed. In case of failure, the error shall be reported as specified in [i.2].

The procedures in the following clauses assume blocking requests as described in clause 8.2.2.

10.1.1 CREATE (C)

The CREATE procedure shall be used by an Originator CSE or AE to create a resource on a Receiver CSE (also called the hosting CSE). The description of CREATE procedure has been divided in two separate clauses, since there is a need to distinguish between Registration related Create and Non-Registration related Create procedures.

The Registration related Create procedure is applicable for the following resource types only:

- <AE>; and
- <remoteCSE>.

Whereas non-registration related Create procedure is applicable for all other resource types described in clause 9.6.

10.1.1.1 Non-registration related CREATE procedure

This procedure is valid for all resources which are not related to registration.

Originator requests to create a resource by using the CREATE method. See clause 8.1.2 for the parameters to be included in the Request message.

Receiver If the request is allowed by the given privileges, the Receiver shall create the resource.

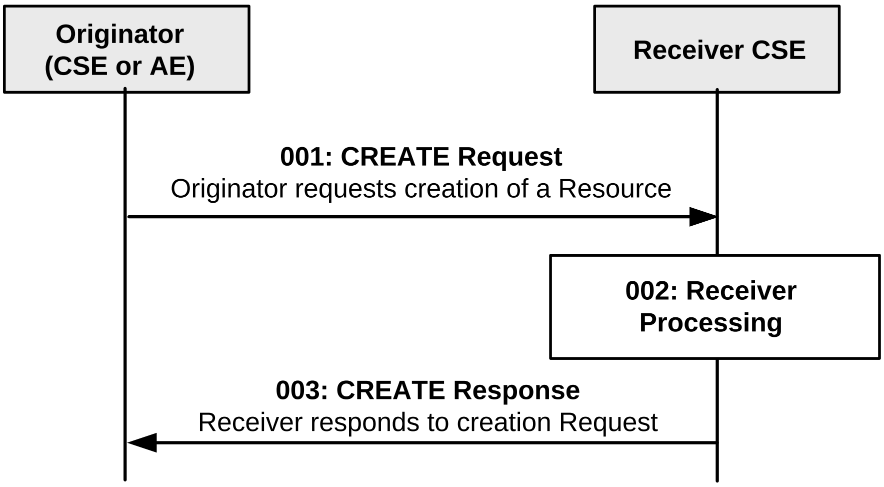


Figure 10.1.1.1-1: Procedure for CREATEing a Resource

Step 001: The Originator shall send the following parameters in the CREATE Request message:

***op*:** C (Create);

***to*:** URI of the target resource where the new resource should be created (parent resource);

***fr*:** ID of the Originator (either the AE or CSE);

[10.1.1.1.b] Editor's Note: AE or CSE may not have the ID upon its registration request. Alternative information for ***fr*** is FFS.

ty: Type of resource to be created;

nm**:** optional name of the resource to be provided by the Originator, where permitted by the resource type as specified in clause 9.6.

NOTE: Some of the resources defined in clause 9.6 have a predefined name. For these resources the parameter ***nm*** is not applicable.

cn**:** attributes of the resource to be provided by the Originator. Of particular importance for the CREATE Request message is the common attribute *resourceType* (clause 9.6.1) which identifies the type of the resource to be Created.

[10.1.1.1.c] Editor's Note: which parameters are deducted to be mandatory/optional in the request is for FFS. Clause 8.1 should provide a generic mechanism to identify mandatory params for a request by looking at how the resource is defined. E.g. If the attribute in the resource definition is identified as mandatory and it is of type RW, then it is also intended as mandatory in the request.

[10.1.1.1.d] Editor's Note: references to elements defined in sect 8 have been added here. They may be removed if we decide to move sect 8.1.2-4 to stage 3 doc. In this case, these references will be removed and sect 1 of this document would replace sect 8.1.2-4 in the TS-0001.

Step 002: The Receiver shall:

1) Check if the Originator has the appropriate privileges for performing the request. Privileges are part of the attribute *accessControlPolicyIDs* of the targeted resource. In case that such an attribute does not exist, the Receiver shall check the *accessControlPolicyIDs* of the parent resource.

[10.1.1.1.e] Editor's Note: how far should the Receiver go back to a parent to look for an accessControlPolicyIDs is FFS

2) Verify that the suggested ***nm***, if provided by the Originator in the Create Request message, does not already exist among child resources of the target resource. If ***nm*** is not provided by the Originator, assign ***nm***.

3) Assign a resource identifier (see *resourceID* attribute in common attribute table 9.6.1-1) to the resource to be created.

4) Assign/modify default values for certain mandatory attributes of the resource, where allowed by the resource itself and if not provided by the Originator itself.

5) the Receiver shall assign a value to the following common attributes specified in clause 9.6.1:

   a) *parentID*;

   b) *creationTime*;

   c) *expirationTime*: if not provided by the Originator, the Receiver shall assign the maximum value possible (within the restriction of the Receiver policies). If the value provided by the Originator cannot be supported, due to either policy or subscription restrictions, the Receiver will assign a new value.

   d) *lastModifiedTime*: which is equals to the creationTime;

   e) Any other RO (Read Only) attributes within the restriction of the Receiver policies.

6) On successful validation of the Create Request, the Receiver shall create the requested resource.

Step 003: The Receiver shall respond with a Response message that shall contain the following parameters:

to**:** Optional. ID of the Originator. In case this is a response carrying the result of an operation triggered by a non-blocking request , ***rd*** information can be used.

fr**:** Optional. ID of the Receiver.

cn**:** URI and optionally the content of the created resource. The Receiver shall provide the content if the created resource contains attributes which were modified by the Receiver in step 002.

See clauses 8.1.3 and 8.1.4 for the parameters to be included in the Response message.

General Exceptions:

1) The Originator does not have the privileges to create a resource on the Receiver. The Receiver responds with an error.

2) The resource with the specified name (if provided) already exists at the Receiver. The Receiver responds with an error.

3) The provided information in ***cn*** is not accepted by the Receiver (e.g. missing mandatory parameter). The Receiver responds with an error.

10.1.1.2 Registration related CREATE procedure

This clause describes the CREATE procedure for <remoteCSE> and <AE> resource type.

10.1.1.2.1 CSE Registration procedure

The procedure for CSE Registration follows the procedure described in clause 10.1.1.1, but with some deviations. Below is the detailed description on how to perform the CSE Registration and which part of the procedure deviates from the one described in clause 10.1.1.1.

The Registration procedure requires the creation of two resources (a <remoteCSE> on the Receiver CSE and a <remoteCSE> on the Originator CSE) rather than one resource. The Registration procedure is always initiated by a CSE in the field domain except in the inter-domain case described in clause 6.3.

Originator: The Originator shall be the registering CSE.

Receiver: The Receiver shall create the <remoteCSE> resource.

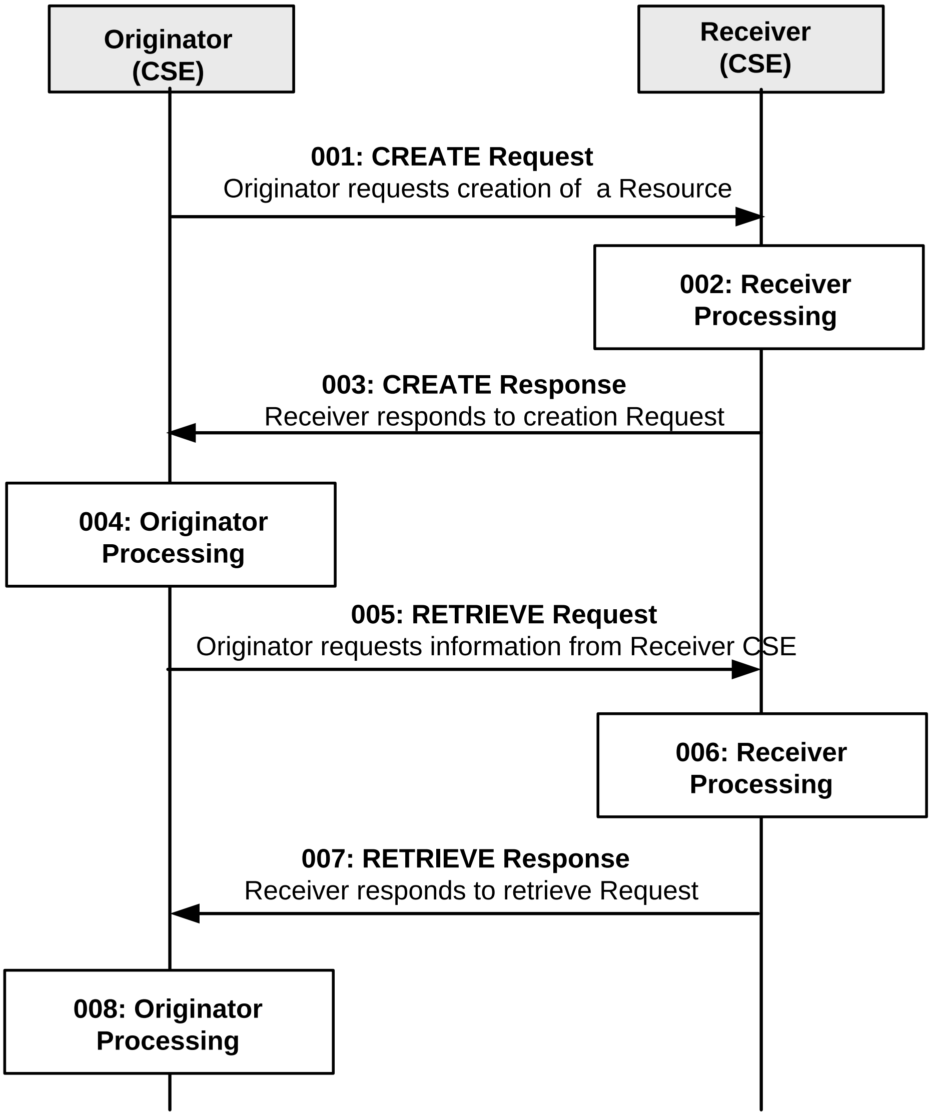

Originator (CSE)
Receiver (CSE)
001: CREATE Request
Originator requests creation of a Resource
002: Receiver Processing
003: CREATE Response
Receiver responds to creation Request
004: Originator Processing
005: RETRIEVE Request
Originator requests information from Receiver CSE
006: Receiver Processing
007: RETRIEVE Response
Receiver responds to retrieve Request
008: Originator Processing

Figure 10.1.1.2.1-1: Procedure for CREATEing a <remoteCSE> Resource

Rappoterur Note: The picture designed for CSE is referenced also for AE in 10.1.1.2, but for AE is incorrect. Duplicate and correct it in 10.1.1.2 ?

Rapporteur Observation: The above stated Rapporteur Note added via contribution ARC-2014-1427. We need to have a contribution to correct the picture for clause 10.1.1.2. Fix the associated descriptions as well.

All the parameters of the request and steps that are not indicated do not deviate from clause 10.1.1.1.

Step 001: The Originator shall send the following information in the CREATE Request message:

***fr*:** CSE-ID.

***cn*:** Attributes of the resource to be provided by the Originator. Of particular importance for the CREATE Request message is the common attribute *resourceType* from clause 9.6.1, which identifies the type of resource; and the attributes *CSE-ID* and *CSEBase*. The *CSEBase* shall contain the absolute URI of the <CSEBase> resource at the Receiver.

Step 002: The Receiver shall:

1) This step from 10.1.1.1 cannot be performed for the creation of <remoteCSE> resource.

All the other steps: 2-6, from step 002 from clause 10.1.1.1 are still applicable.

NOTE: Optionally, if the M2M Service Provider supports inter-domain communication, the Receiver could perform this step if the attribute *CSEBase* (part of the ***cn*** parameter of the request) contains the public domain of the CSE. The Receiver could construct the domain as described in clause 6.4 and 6.5. The Receiver could add an AAA or AAAA record in DNS with the public domain name of the Originator CSE and the IP address of the IN-CSE associated with the Originator.

Step 003: See clause 10.1.1.1.

Step 004: The Originator, upon receipt of the CREATE response message, shall create a <remoteCSE> resource locally under its <CSEBase> resource. This resource is representing the Receiver CSE. The Originator shall provide the appropriate values to all mandatory parameters as described in clause 9.6.4.

Step 005: The Originator may issue a RETRIEVE Request towards the Receiver (same ***to*** as for the CREATE request message) to obtain the optional parameters of the <remoteCSE> resource created at the Receiver as for step 004 (e.g. *labels*, *accessControlPolicyIDs* attributes). The RETRIEVE procedure is described in clause 10.1.2.

See clauses 8.1.2 for the information to be included in the Request message.

Step 006: The Receiver verifies that the Originator has the appropriate privileges to access the information.

Step 007: The Receiver sends a RETRIEVE response message, according to the procedure described in clause 10.1.2.

See clauses 8.1.3 and 8.1.4 for the information to be included in the Response message.

Step 008: The Originator shall update the created <remoteCSE> resource for the Receiver with the information obtained in step 007.

General Exceptions:

All exceptions from clause 10.1.1.1 are applicable; in addition the following exception may occur:

1) The Originator does not have the privileges to retrieve the attributes of the Receiver CSE. The Receiver responds with an error.

10.1.1.2.2 Application Entity Registration procedure

The procedure for AE registration follows the procedure described in clause 10.1.1.1 with the following exception:

Originator: The Originator shall be the registering AE.

Receiver: The Receiver shall allow the creation of the <AE> resource according to the access control policy. The Receiver shall use, if present, the suggested name from the information parameter ***nm*** in the request. If the suggested name of the resource cannot be used (i.e. a resource with that name already exists) the Receiver shall reject the CREATE Request. If the request does not provide the optional ***nm***, then the Receiver shall create the resource and assign a name to it.

Step 001: The Originator shall send the following information in the CREATE Request message:

fr**:** AE-ID.

Step 002: for the <AE> resource the Receiver cannot perform the action described in the first step of clause 10.1.1.1.

All the other parameters of the request and the steps that follow do not deviate from clause 10.1.1.1.

10.1.2 RETRIEVE (R)

The RETRIEVE operation shall be used for retrieving the information stored for any of the attributes for a resource at the Receiver CSE. The Originator CSE or AE may request to retrieve a specific attribute by including the name of such attribute in the ***cn*** parameter in the request message.

Originator requests retrieval of all attributes or a specific attributes of the target resource by using RETRIEVE Request. See clause 8.1.2 for the information to be included in the Request message. If only some specific attributes need to be retrieved, the name of such attributes shall be included in the ***cn*** parameter of the Request message.

Receiver 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, otherwise an error indication shall be returned.

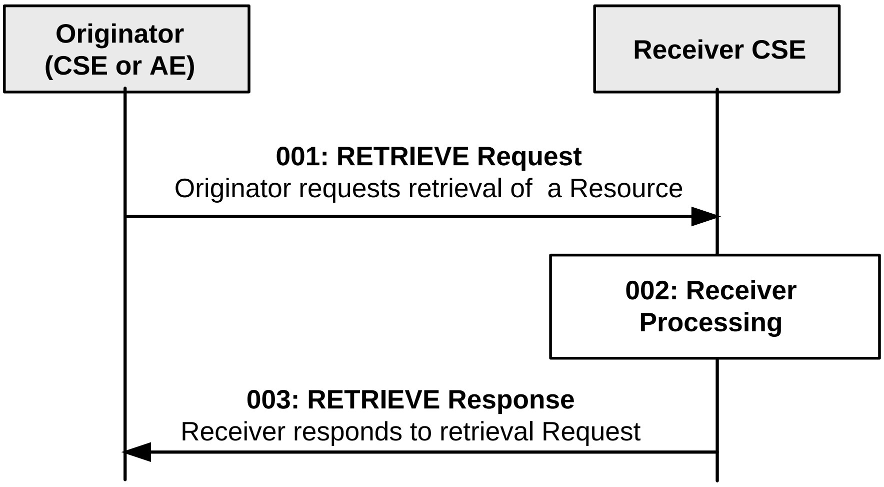


Figure 10.1.2-1: Procedure for RETRIEVing a Resource

Step 001: The Originator shall request to RETRIEVE a resource or a specific attribute within a resource at the Receiver.

***op*:** R (Retrieve).

***to*:** URI of the target resource or the target attribute.

***fr*:** ID of the Originator (either the AE or CSE).

cn: Optional. If included, ***cn*** includes the name of the attributes that need to be retrieved.

Step 002: The Receiver shall verify the existence of the target resource or the attribute and check if the Originator has appropriate privileges to retrieve information stored in the resource/attribute.

Step 003: The Receiver shall respond with a Response message that shall contain the following information:

***to*:** Optional. ID of the Originator.

***fr*:** Optional. ID of the Receiver.

***cn*:** content of resource/attribute retrieved.

General Exceptions:

1) The targeted resource/attribute in ***to*** parameter does not exist. The Receiver responds with an error.

2) The Originator does not have privileges to retrieve information stored in the resource on the Receiver. The Receiver responds with an error.

10.1.3 UPDATE (U)

The UPDATE operation shall be used for updating the information stored for any of the attributes at a target resource. Especially important is the *expirationTime*, since a failure in refreshing this attribute may result in the deletion of the resource. The Originator CSE or AE can request to update, create or delete specific attribute(s) at the target resource by including the name of such attribute(s) and its values in the request message.

Originator requests update any of the attributes at the target resource by using UPDATE Request message. The Originator shall send new (proposed) values for the attribute(s) that need to be updated. The UPDATE operation allows to modify the attributes (defined in clause 9.6) and that are indicated as "RW" (Read Write) for the specific resource type.

The **Originator** requests to create attributes at the target resource by using UPDATE Request message. The Originator shall send the name of the attributes to be created (defined in clause 9.6) that are indicated as "RW" (Read Write) for the specific resource type and their associated values in the Request message,

The **Originator** requests to delete attributes at the target resource by using UPDATE Request message. The Originator shall send the name of the attributes to be deleted (defined in clause 9.6) for the specific resource type with their value set to NULL, in the Request message,

See clause 8.1.2 for the information to be included in the Request message.

NOTE: Update operation can also be used for Execute operation. Such use of the Update operation does not use ***cn*** parameter.

Receiver The Receiver verifies the existence of the addressed resource, the validity of the attributes provided and the privileges to modify them, shall update the attributes provided and shall return a Response message to the Originator with the operation results as specified in clause 8.1.3.

If the attributes provided do not exist, after verifying the existence of the addressed resource, the Receiver validates the attributes provided and the privileges to create them. On successful validation, the Receiver shall create the attributes provided with their associated values and shall return a Response message to the Originator with the operation results as specified in clause 8.1.3.

If the attributes provided have their value set to NULL, after verifying the existence of the addressed resource, the Receiver validates the attributes provided and the privileges to delete them. On successful validation, the Receiver shall delete such attributes and shall return a Response message to the Originator with the operation results as specified in clause 8.1.3.

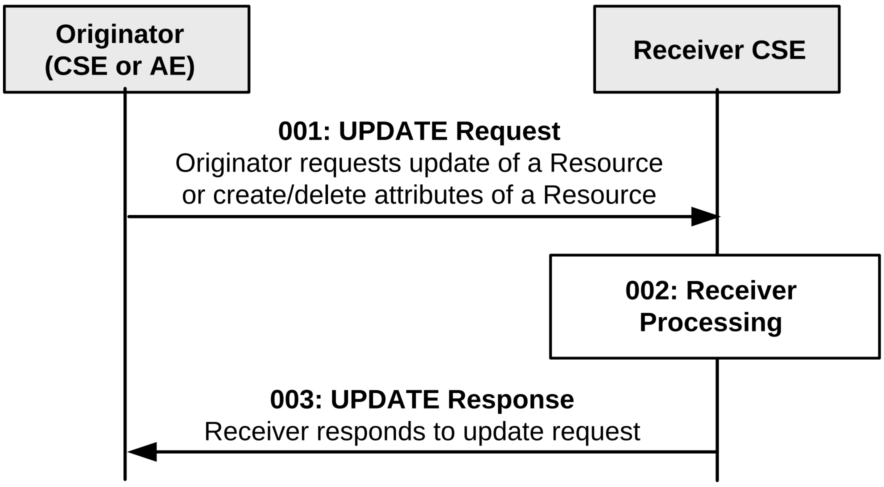

Originator
(CSE or AE)
Receiver CSE
001: UPDATE Request
Originator requests update of a Resource
or create/delete attributes of a Resource
002: Receiver
Processing
003: UPDATE Response
Receiver responds to update request

Figure 10.1.3-1: Procedure for UPDATing a Resource

Step 001: The Originator shall send the following parameters in the UPDATE Request message:

***op*:** U (Update).

***to*:** URI to the target resource.

***fr*:** ID of the Originator (either the AE or CSE).

cn*:** Information related to the attribute(s) to be updated, created or deleted at the target resource. The name of such attribute(s) and associated updated or assigned values in the ***cn parameter. For the Update operation used for Execute device management operation, ***cn*** parameter does not exist.

Step 002: The Receiver shall validate if the Originator has appropriate privileges to perform the modification to the target resource. On successful validation, the Receiver shall update the resource as requested. If the attributes provided do not exist, the Receiver shall validate if the Originator has appropriate privileges to create the attributes at the target resource. On successful validation, the Receiver shall create the attributes with their associated values at the resource as requested. If the attributes provided have their value set to NULL, the Receiver shall validate if the Originator has appropriate privileges to delete the attributes at the target resource. On successful validation, the Receiver shall delete such attributes.

Step 003: The Receiver shall respond with a Response message that shall contain the following parameters:

***to*:** Optional. ID of the Originator.

***fr*:** Optional. ID of the Receiver.

cn: Optional. Content replaced, created or deleted.

***rs*:** Operation result.

General Exceptions:

1) The targeted resource in ***to*** parameter does not exist. The Receiver responds with an error.

2) The Originator does not have the privilege to modify the resource, create attributes or delete attributes on the Receiver. The Receiver responds with error.

3) The provided information in the ***cn*** is not accepted by the Receiver. The Receiver responds with error.

10.1.4 DELETE (D)

The DELETE operation shall be used by an Originator CSE or AE to delete a resource on a Receiver CSE (also called the hosting CSE). The description of DELETE procedure has been divided in two separate clauses, since there is a need to distinguish between Deregistration related Delete and Non-Deregistration related Delete procedures.

The Deregistration related Delete procedure is applicable for the following resource types only:

- <AE>, and
- <remoteCSE>

10.1.4.1 Non-deregistration related DELETE procedure

This procedure is valid for all resources which are not related to deregistration.

The DELETE operation shall be used by an Originator CSE or AE to delete a resource at a Receiver CSE. For such operation, the DELETE procedure shall consist of the deletion of all related information of the target resource.

Originator requests deletion of a resource by using a DELETE Request message. See clause 8.1.2 for the information to be included in the Request message.

Receiver The Receiver verifies the existence of the requested resource, and the privileges for deleting the resource.

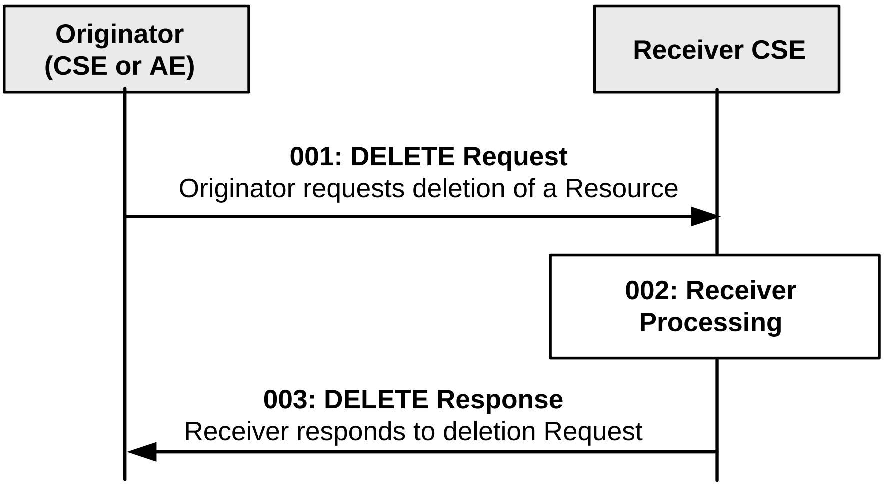


Figure 10.1.4-1: Procedure for DELETING a Resource

Step 001: The Originator shall send a DELETE Request message to the Receiver.

***op*:** D (Delete).

***to*:** URI of the target resource.

***fr*:** ID of the Originator (either the AE or CSE).

Step 002: The Receiver shall verify the existence of the requested resource and if the Originator has the appropriate privilege to delete the resource. On successful validation, the Receiver shall check for child resources and delete all child resources and the associated references in parent resources and it shall remove the resource itself.

Step 003: The Receiver shall respond with a Response message that shall contain the following information:

to*:** Optional. ID of the Originator. In case this is a response carrying the result of an operation triggered by a non-blocking request, ***rd information can be used.

***fr*:** Optional. ID of the Receiver.

***rs*:** Operation result.

General Exceptions:

1) The targeted resource in ***to*** information does not exist. The Receiver responds with an error.

2) The Originator does not have the privileges to delete the resource on the Receiver. The Receiver responds with an error.

10.1.4.2 Deregistration related DELETE procedure

This clause describes the CREATE procedure for <remoteCSE> and <AE> resource type.

10.1.4.2.1 CSE Deregistration procedure

The procedure for CSE Deregistration follows the procedure described in clause 10.1.4.1, but with some exceptions. Below is the detailed description on how to perform the CSE Deregistration and which part of the procedure deviates from the one described in clause 10.1.4.1.

The Deregistration procedure accompanies the deletion of two resources (a <remoteCSE> on the hosting CSE and a <remoteCSE> on the Originator CSE) rather than one resource. The Deregistration procedure can be initiated by either Registree CSE or Registrar CSE.

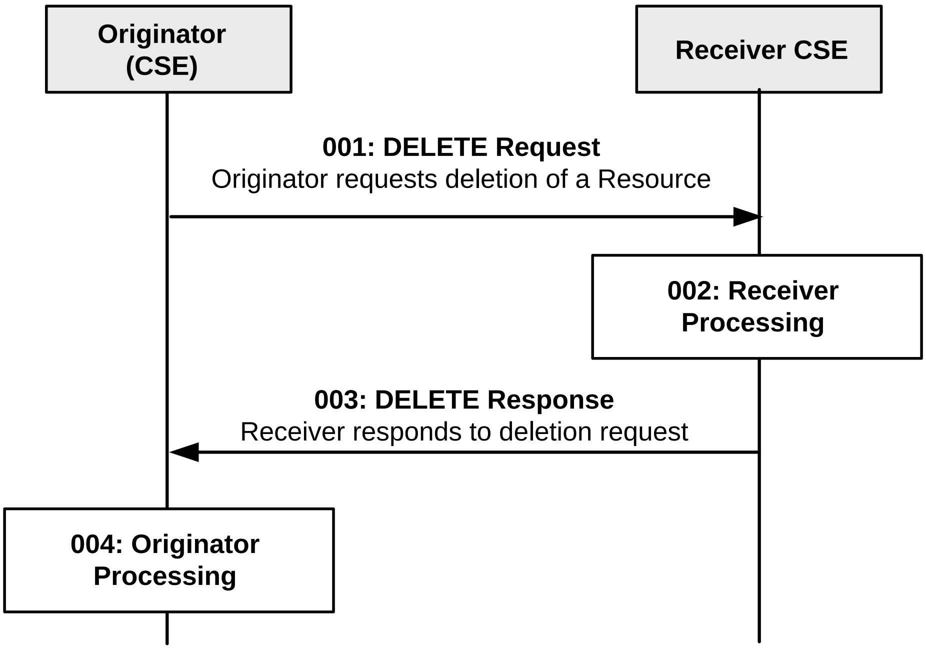

Originator (CSE)
Receiver CSE
001: DELETE Request
Originator requests deletion of a Resource
002: Receiver Processing
003: DELETE Response
Receiver responds to deletion request
004: Originator Processing

Figure 10.1.4.2.1-1: Procedure for DELETING a <remoteCSE> Resource

Step 001: See clause 10.1.4.1.

Step 002: See clause 10.1.4.1.

Step 003: See clause 10.1.4.1.

Step 004: The Originator, upon receipt of the DELETE response, shall delete a <remoteCSE> resource locally under its <CSEBase> resource.

General Exceptions:

All exceptions from 10.1.4.1 are applicable; in addition the following exception may occur:

1) If the Receiver rejects the DELETE request and responds with an error in the DELETE response, the Originator cannot perform the action described in the Step 004.

10.1.4.2.2 Application Entity Deregistration procedure

The procedure for AE Deregistration follows the procedure described in clause 10.1.4.1.

10.1.5 NOTIFY (N)

The NOTIFY operation shall be used for notifying information.

Originator: The Originator requests to notify an entity by using NOTIFY method. See clause 8.1.2 for the information to be included in a Request message.

Receiver: The Receiver responds to the Originator with the operation results as specified in clause 8.1.3.

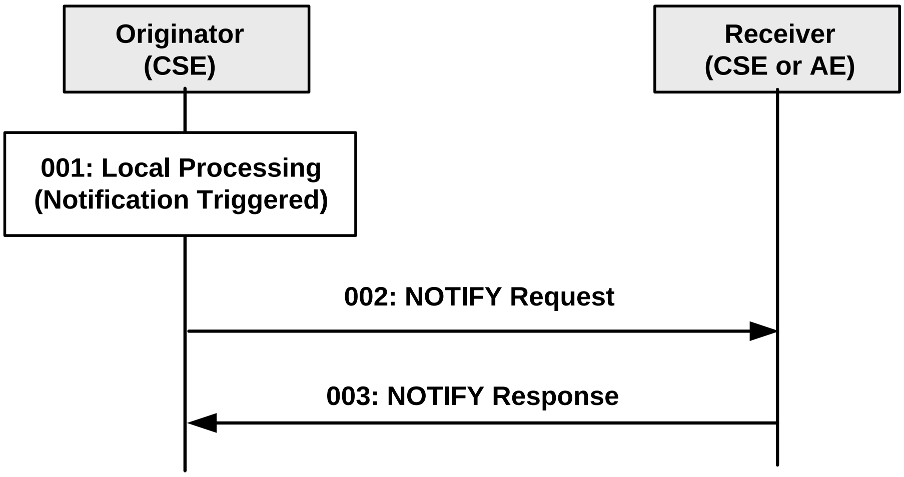


Figure 10.1.5-1: Procedure for NOTIFYing Information

[10.1.5.a] Editor's Note: not sure how to align this procedure with the other ones. The Receiver should also have a local processing, but if the address of the resource in the to is not one of the defined resource types what the receiver does is out of scope. However if the NOTIFY will corresponds to a create instance then the procedures described in the create should apply. This operation needs more refining.

Step 001: A notification to be sent to the Receiver is triggered in the Originator.

Step 002: The Originator shall send the following parameters in the NOTIFY Request message:

***op*:** N (Notify).

***to*:** URI where the notification should be sent to.

***fr*:** ID of the Originator (only a CSE).

***cn*:** Notification data and/or Notification reference.

Step 003: The Receiver responds with a Response message that shall contain the following information:

***rs*:** operation result.

General Exceptions:

1) None identified.

[10.1.5.b] Editor's note: General exceptions are FFS in particular for notification failures.

10.2 Resource Type-Specific Procedures

The basic procedure for the corresponding operations as specified in clause 10.1 shall be performed with the modifications specific to the resource type procedures as described in clause 10.2.

For resources without defined resource type-specific operations, the basic operations in clause 10.1 shall apply.

NOTE: Where the procedures in clause 10.2 conflict with the procedures in clause 10.2. the procedures in clause 10.2 take precedence.

10.2.1 <AE> Resource

10.2.1.1 Create <AE>, aka Application Registration

This flow is used for creating an <AE> resource. This operation is part of the registration procedure for AEs on the Registrar CSE (which is also the hosting CSE), as described in clause 10.1.1.2.2.

The Create procedure shall be according to table 10.2.1.1-1.

Table 10.2.1.1-1 <AE> CREATE

<AE> CREATE	
Associated Reference Point	Mca only
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***fr***: AE only **cn**: The resource content shall provide the information as defined in clause 9.6.5.
Pre-Processing at Originator	According to clause 10.1.1.2.2
Processing at Receiver	According to clause 10.1.1.2.2
Information on Response message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: URI of the created <AE> resource, according to clause 10.1.1
Post-Processing at Originator	According to clause 10.1.1.2.2
Exceptions	According to clause 10.1.1.2.2

10.2.1.2 Retrieve <AE>

This flow is used for retrieving the representation of the <AE> resource with its attributes.

Table 10.2.1.2-1 <AE> RETRIEVE

<AE> RETRIEVE	
Associated Reference Point	Mca, Mcc, Mcc’
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: void
Pre-Processing at Originator	According to clause 10.1.2
Processing at Receiver	According to clause 10.1.2
Information on Response message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: attributes of the <AE> resource as defined in clause 9.6.5.
Post-Processing at Originator	According to clause 10.1.2
Exceptions	According to clause 10.1.2

10.2.1.3 Update <AE>

This flow is used for updating the attributes and the actual data of an <AE> resource.

Table 10.2.1.3-1 <AE> UPDATE

<AE> UPDATE	
Associated Reference Point	Mca, Mcc, Mcc’
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: attributes of the <AE> resource as defined in 9.6.5 which need be updated,
Pre-Processing at Originator	According to clause 10.1.3
Processing at Receiver	According to clause 10.1.3
Information on Response message	According to clause 10.1.3
Post-Processing at Originator	According to clause 10.1.3
Exceptions	According to clause 10.1.3

10.2.1.4 Delete <AE>

This flow is used for deleting the <AE> resource with all related information.

Table 10.2.1.4-1 <AE> DELETE flow

<AE> DELETE	
Associated Reference Point	Mca, Mcc, Mcc’
Information on Request message	All parameters defined in table 8.1.2.1-1 apply
Pre-Processing at Originator	According to clause 10.1.4
Processing at Receiver	According to clause 10.1.4
Information on Response message	According to clause 10.1.4
Post-Processing at Originator	According to clause 10.1.4
Exceptions	According to clause 10.1.4

10.2.2 <remoteCSE> Resource

10.2.2.1 Create <remoteCSE>, aka CSE Registration

This flow is used for creating a <remoteCSE> resource. It is part of the registration procedure for remote CSEs on the Registrar CSE (which is also the hosting CSE), as described in clause 10.1.1.2.1.

Table 10.2.2.1-1 <remoteCSE> CREATE

<remoteCSE> CREATE	
Associated Reference Point	Mcc and Mcc’ only
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***fr***: CSE only **cn**: The resource content shall provide the information as defined in clause 9.6.4.
Pre-Processing at Originator	According to clause 10.1.1.2.1
Processing at Receiver	According to clause 10.1.1.2.1 A remote CSE resource is created If the IN-CSE is the receiver and if the M2M SP policies do allow access to the CSEs across multiple domains, then the IN shall create the appropriate entry in the M2M SP's DNS for successfully registered CSE.
Information on Response message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: URI of the created <CSE> resource, according to clause 10.1.1.2.1
Post-Processing at Originator	According to clause 10.1.1.2.1. the originator starts a Retrieve operation and uses the result to create a remoteCSE representation of the Receiver
Exceptions	According to clause 10.1.1.2.1

10.2.2.2 Retrieve <remoteCSE>

This flow is used for retrieving the representation of the <remoteCSE> resource with its attributes.

Table 10.2.2.2-1 <remoteCSE> RETRIEVE

<remoteCSE> RETRIEVE	
Associated Reference Point	Mca, Mcc, Mcc'
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: void
Pre-Processing at Originator	According to clause 10.1.2
Processing at Receiver	According to clause 10.1.2
Information on Response message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: attributes of the <remoteCSE> resource as defined in clause 9.6.4
Post-Processing at Originator	According to clause 10.1.2.
Exceptions	According to clause 10.1.2

10.2.2.3 Update <remoteCSE>

This flow is used for updating the attributes and the actual data of an <remoteCSE> resource.

Table 10.2.2.3-1 <remoteCSE> UPDATE

<remoteCSE> UPDATE	
Associated Reference Point	Mcc, Mcc'
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: attributes of the <remoteCSE> resource as defined in 9.6.4 which need be updated,
Pre-Processing at Originator	According to clause 10.1.3
Processing at Receiver	According to clause 10.1.3
Information on Response message	According to clause 10.1.3
Post-Processing at Originator	According to clause 10.1.3
Exceptions	According to clause 10.1.3

10.2.2.4 Delete <remoteCSE>

This flow is used for deleting the <remoteCSE> resource with all related information.

If the IN-CSE is the receiver and it has created an entry in the DNS to allow access to the CSE across multiple M2M domains, then it shall delete the entry from the DNS.

Table 10.2.2.4-1 <remoteCSE> DELETE

<remoteCSE> DELETE	
Associated Reference Point	Mca, Mcc, Mcc’
Information on Request message	All parameters defined in table 8.1.2.1-1 apply
Pre-Processing at Originator	According to clause 10.1.4
Processing at Receiver	According to clause 10.1.4 If the IN-CSE is the receiver and it has created an entry in the DNS to allow access to the CSE across multiple M2M domains, then it shall delete the entry from the DNS.
Information on Response message	According to clause 10.1.4
Post-Processing at Originator	According to clause 10.1.4
Exceptions	According to clause 10.1.4

10.2.3 <CSEBase> Resource

10.2.3.1 Create <CSEBase>

The Create operation shall not apply to <CSE base>. <CSE base> is expected to be created via management operation not define in this version of the specification.

10.2.3.2 Retrieve <CSEBase>

This flow is used for retrieving the representation of the <CSEBase> resource with its attributes, to complete the registration procedure as described in 10.1.1.2.1 (CSE Registration procedure)

Table 10.2.3.2-1 <CSEBase> RETRIEVE

<CSEBase> RETRIEVE	
Associated Reference Point	Mcc, Mcc’
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: void
Pre-Processing at Originator	According to clause 10.1.2 and 10.1.1.2.1
Processing at Receiver	According to clause 10.1.2 and 10.1.1.2.1
Information on Response message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: attributes of the <CSEBase> resource as defined in clause 9.6.3
Post-Processing at Originator	According to clause 10.1.2 and 10.1.1.2.1 A remote CSE resource is created using the retrieved resource
Exceptions	According to clause 10.1.2 and 10.1.1.2.1

10.2.3.3 Update <CSEBase>

The Create operation shall not apply to <CSE base>. <CSE base> is expected to be created via management operation not define in this version of the specification.

10.2.3.4 Delete <CSEBase>

The Delete operation shall not apply to <CSE base>. <CSE base> is expected to be created via management operation not define in this version of the specification.

10.2.4 <container> Resource

10.2.4.1 Create <container>

This flow is used for creating a <container> resource.

Table 10.2.4.1-1: <container> CREATE

<container> CREATE	
Associated Reference Point	MCA, Mcc and Mcc'
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: **cn**: The resource content shall provide the information as defined in clause 9.6.6.
Pre-Processing at Originator	According to clause 10.1.1.1
Processing at Receiver	According to clause 10.1.1.1
Information on Response message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: URI of the created <Container> resource, according to clause 10.1.1.1
Post-Processing at Originator	According to clause 10.1.1.1
Exceptions	According to clause 10.1.1.1

10.2.4.2 Retrieve <container>

This flow is used for retrieving the attributes of a <container> resource.

Table 10.2.4.2-1: <container> RETRIEVE

<container> RETRIEVE	
Associated Reference Point	Mca, Mcc, Mcc'
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: void
Pre-Processing at Originator	According to clause 10.1.2
Processing at Receiver	According to clause 10.1.2
Information on Response message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: attributes of the <remoteCSE> resource as defined in clause 9.6.6
Post-Processing at Originator	According to clause 10.1.2.
Exceptions	According to clause 10.1.2

10.2.4.3 Update <container>

This flow is used for updating the attributes and the actual data of a <container> resource.

Table 10.2.4.3-1: <container> UPDATE

<container> UPDATE	
Associated Reference Point	Mca, Mcc, Mcc’
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: attributes of the <remoteCSE> resource as defined in 9.6.6 which need be updated,
Pre-Processing at Originator	According to clause 10.1.3
Processing at Receiver	According to clause 10.1.3
Information on Response message	According to clause 10.1.3
Post-Processing at Originator	According to clause 10.1.3
Exceptions	According to clause 10.1.3

10.2.4.4 Delete <container>

This flow is used for deleting a <container> resource residing under a <container> resource.

Table 10.2.4.4-1: <container> DELETE

<remoteCSE> DELETE	
Associated Reference Point	Mca, Mcc, Mcc’
Information on Request message	All parameters defined in table 8.1.2.1-1 apply
Pre-Processing at Originator	According to clause 10.1.4
Processing at Receiver	According to clause 10.1.4 with thefollowing If the IN-CSE is the receiver and it has created an entry in the DNS to allow access to the CSE across multiple M2M domains, then it shall delete the entry from the DNS.
Information on Response message	According to clause 10.1.4
Post-Processing at Originator	According to clause 10.1.4
Exceptions	According to clause 10.1.4

10.2.5 Access to Remotely Hosted Resources via <delivery>

10.2.5.1 Introduction to usage of <delivery> resource type

In this introduction an example for delivering information from a source CSE to a target CSE via the use of the <delivery> resource is explained.

The information flow depicted in figure 10.2.5.1-1 defines the exchange of Requests/Responses for processing an original request targeting a resource that is not hosted on the Registrar CSE of the request Originator. The following assumptions hold:

- Originator is AE1.
- AE1 is registered with CSE1, i.e. CSE1 is the Registrar CSE for AE1.
- The original Request is an UPDATE to a remote resource hosted on CSE3, i.e. CSE3 is the hosting CSE for the target resource.
- UPDATE options in the original Request are selected such that no feedback after completion of the update operation was requested, i.e. AE1 decided that it does not need to hear back from CSE3; this is expressed by setting the ***rc*** information to "nothing", see clause 8.1.2.

- Delivery related parameters included in the original UPDATE request (may be set via CMDH policies): ***rqet***, ***ec***, ***da*** and ***rp***:

  - ***rqet*** indicates how long the forwarding of the request can last at most.

  - ***ec*** indicates the event category that should be used by CMDH to handle this request.

  - ***rp*** indicates how long after the request has expired, the local request context should still be available for retrieving status or result information.

  - ***da*** would be set to ON indicating that <delivery> resource shall be used for forwarding the request.

- CSE1 is the CSE of an Application Service Node.

- CSE1 is registered with CSE2 and interacts with CSE2 via the reference point Mcc(1).

- CSE2 is the CSE of a Middle Node.

- CSE2 is registered with CSE3 and interacts with CSE3 via the reference point Mcc(2)

- CSE3 is the CSE of an Infrastructure Node.

The Originator AE1 shall get a confirmation from CSE1 when the original Request is accepted. The response informs AE1 that CSE1 has accepted the Request and has accepted responsibility to execute on the requested operation. Furthermore, AE1 has expressed by setting ***rc*** to "nothing" that no result of the requested operation is expected to come back from CSE3. With the provided reference (Req-Ref in figure 10.2.5.1-1. AE1 can retrieve the status of the issued request at a later time, for instance to find out if the request was already forwarded to CSE2 or if it is still waiting for being forwarded on CSE1. Before accepting the request from AE1, CSE1 has also verified if the delivery related parameters expressed by AE1 (settings of ***rqet*** and ***ec***) are in line with provisioned CMDH policies. AE1 may not be authorized to use certain values for ***rqet*** or ***ec***.

In line with the delivery related parameters, CSE1 is generating a local <delivery> resource on CSE1 and attempts to forward the content of it in line with provisioned CMDH policies at a suitable time and via a suitable connection to CSE2 by requesting the creation of a <delivery> resource on CSE2. In this example case, the *lifespan* attribute of this delivery resource is set to the same value as the ***rqet*** parameter expressed by AE1. In general - i.e. also in cases where more than one original request is aggregated into a single create request for a <delivery> resource - the *lifespan* and *eventCat* attributes of the created <delivery> resource shall be set consistent with the ***rqet*** and ***ec*** parameters in the set of original requests. See the attribute definitions in clause 9.6.11.

CSE1 shall use a blocking request for requesting creation of a <delivery> resource on CSE2.

When CSE2 has accepted the incoming request from CSE1, CSE1 may delete the *data* attribute of the local <delivery> resource. Furthermore - if the expiration time of the local <delivery> resource is not exhausted - the Registrar CSE shall update the status of the local <delivery> resource to indicate that it has been forwarded to CSE2. CSE1 shall also update the status of the original request to indicate that it has been forwarded and it may delete the *data* attribute of the original request.

When CSE2 has accepted the request to create a local <delivery> resource, it shall attempt to forward it to CSE3. In line with the delivery related parameters, CSE2 shall create a local <delivery> resource on CSE2 and shall attempt to forward it in line with provisioned CMDH policies at a suitable time and via a suitable connection to CSE3 by requesting the creation of a <delivery> resource on CSE3.

CSE2 shall use a blocking request for requesting creation of a <delivery> resource on CSE3.

When CSE3 has accepted the incoming request from CSE2, CSE2 may delete the *data* attribute of the local <delivery> resource. Furthermore - if the expiration time of the local <delivery> resource is not exhausted - the Registrar CSE shall update the status of the local <delivery> resource to indicate that it has been forwarded to CSE3.

When CSE3 has accepted the request to create a local <delivery> resource, it shall determine that the target of the delivery was CSE3 itself. Therefore it shall forward internally the original request contained in the *data* attribute of the <delivery> resource.

Within CSE3, functions that are responsible for checking and executing local access to resources in CSE3 will execute the originally requested UPDATE operation. If successful, the targeted resource will be updated with the content provided by the Originator.

Since in the depicted case no result needed to be sent back to the Originator, the processing for the requested operation is then completed.

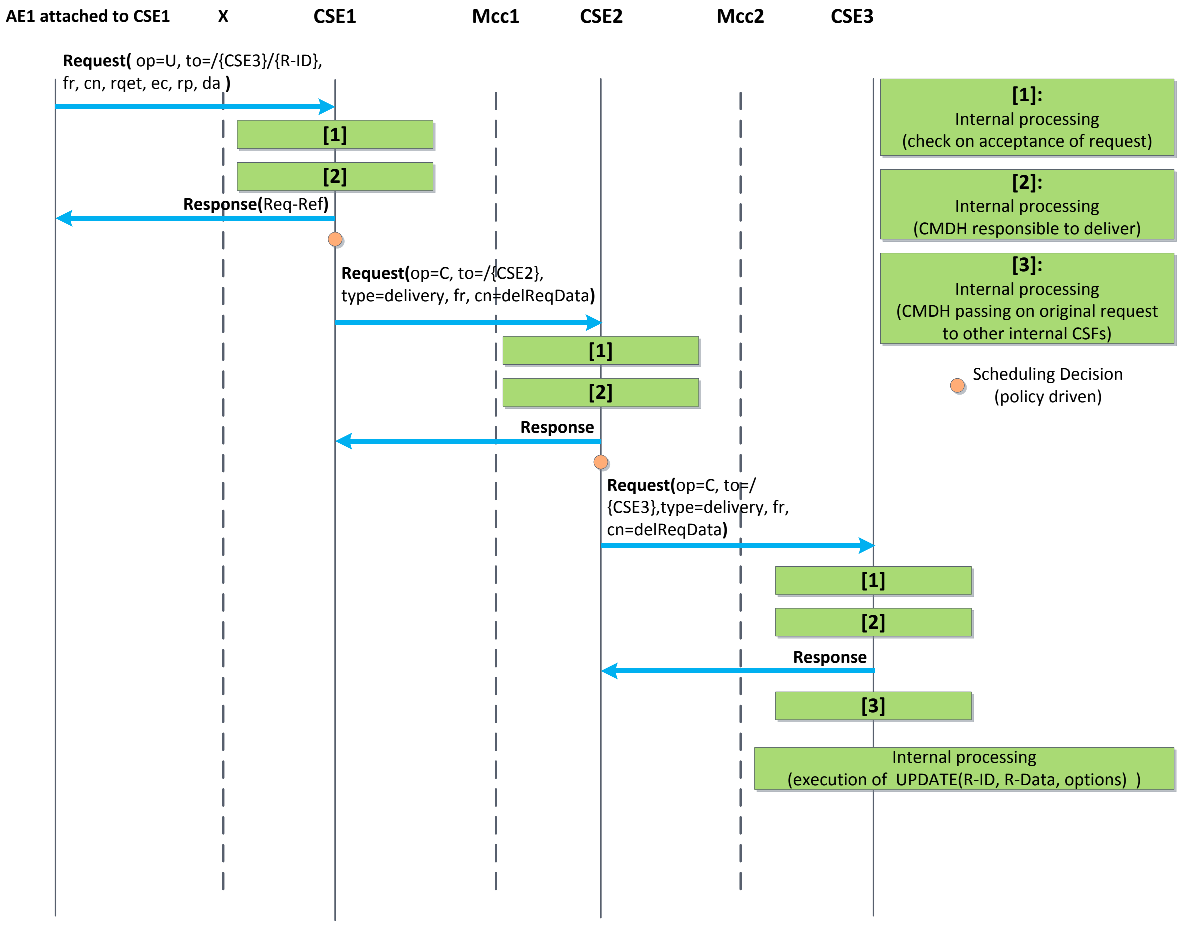

AE1 attached to CSE1
X
CSE1
Mcc1
CSE2
Mcc2
CSE3
Request(op=U, to=/{CSE3}/{R-ID}, fr, cn, rqet, ec, rp, da)
[1]
[2]
Response(Req-Ref)
Request(op=C, to=/{CSE2}, type=delivery, fr, cn=delReqData)
[1]
[2]
Response
Request(op=C, to=/ {CSE3},type=delivery, fr, cn=delReqData)
[1]
[2]
Response
[3]
Internal processing (execution of UPDATE(R-ID, R-Data, options))
[1]: Internal processing (check on acceptance of request)
[2]: Internal processing (CMDH responsible to deliver)
[3]: Internal processing (CMDH passing on original request to other internal CSFs)
Scheduling Decision (policy driven)

Figure 10.2.5.1-1: CMDH information flow for 2 hops - no result needs to be returned after operation completes

The following procedures shall be triggered by requesting the corresponding operations on a <delivery> resource:

- Initiate the delivery of one or more original request(s) stored for later forwarding from one CSE to another CSE:
  - Request a CREATE operation for a <delivery> resource from an issuing CSE to a receiving CSE.
  - The original request(s) need to be contained in the "*data*" attribute of the <delivery> resource.
  - If successful, the receiving CSE takes the responsibility to further execute on the delivery process for the original Request.
  - If not successful, the issuing CSE cannot assume that the receiving CSE will carry out the delivery of the original request.
- Get information about the status of a pending delivery process for an original request:
  - Request a RETRIEVE operation of the content of a <delivery> resource representing a pending delivery or part of it.
  - The status of the pending forwarding process is reflected the "*deliveryMetaData*" attribute defined in the <delivery> resource.
- Change parameters of pending delivery process:

  - Request an UPDATE operation on applicable attributes of the <delivery> resource representing the pending delivery.

  - For instance the time allowed for completion of a delivery process could be modified by updating the "*lifespan*" attribute of an existing <delivery> resource.

- Cancel a pending delivery request:

  - Request a DELETE operation of a <delivery> resource that represents a pending delivery process.

10.2.5.2 Create <delivery>

This procedure shall be used for requesting a CSE to take responsibility to deliver the provided data to a target CSE in line with CMDH parameters and provisioned CMDH policies in case <delivery> resource based CMDH processing is used. If indicated by the Originator, the Receiver shall confirm the acceptance of delivery responsibility by a successful Response.

Originator: The Originator of a Create request for a <delivery> resource can only be a CSE. The Originator needs to provide the content of a <delivery> resource type together with the Create request or can Update it after a successful creation of the <delivery> resource with empty *data* attribute. Otherwise the Receiver cannot accept the Create Request. The Originator shall use a blocking request for issuing the Create request to the Receiver.

Receiver: The receiver of a Create request for a <delivery> resource is a Registrar CSE or the Originator and it shall check the access control policies to assure the Originator is authorized to request a delivery procedure. The Receiver of the Create Request shall further check whether the provided attributes of the <delivery> resource that is requested to be created represents a valid request for forwarding data to a target CSE. If the Originator of the Create request is authorized and the Request is valid, the Receiver shall check whether it can actually satisfy the requested delivery in line with provisioned CMDH policies and requested *eventCat* and *lifespan* attributes of the <delivery> resource. If all these checks are positive, the Receiver shall create the requested <delivery> resource and assumes responsibility for delivering the requested data to the target CSE as soon as the content of the *data* attribute is available. In case an operation result is expected by the Originator, the Receiver shall confirm acceptance of the responsibility by indicating a successful creation of the <delivery> resource. If the Receiver CSE is the target CSE of the requested delivery, it shall forward the content of the delivered data - which represents one or more forwarded original request(s) - to the internal functions that handle incoming requests and continue processing of the forwarded request(s).

Table 10.2.5.2-1: <delivery> CREATE

<delivery> CREATE	
Associated Reference Point	Mcc, Mcc’
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: **fr:** CSE only **cn**: The resource content shall provide the information as defined in clause 9.6.11 ***rt*****:** Shall be set to "blockingRequest" which means a blocking request is issued
Pre-Processing at Originator	According to clause 10.1.1.1 with the following specific processing: . The Originator needs to provide the content of a <delivery> resource type together with the Create request or can Update it after a successful creation of the <delivery> resource with empty *data* attribute. Otherwise the Receiver cannot accept the Create Request. The Originator shall use a blocking request for issuing the Create request to the Receiver.
Processing at Receiver	According to clause 10.1.1.1 with the following specific processing: • Check whether the provided attributes of the <delivery> resource that is requested to be created represents a valid request for delivering data to a target CSE. • Check whether Receiver CSE can actually satisfy the requested delivery in line with provisioned policies and requested delivery parameters • If all checks are positive, the receiver shall create the requested <delivery> resource and assumes responsibility for delivering the provided data to the target CSE. • If the Receiver CSE is the target CSE of the requested delivery, it shall forward the content of the delivered data attribute to the internal CSFs that will interpret the delivered data as a forwarded request(s) from a remote Originator.
Information on Response message	All parameters defined in table 8.1.2.1-1 apply, with the following specific information: In case the Originator CSE has not asked for a Result of the requested Operation (***rc*** set to "nothing"), the Response only contains an Acknowledgement indicator. This only indicates that the Receiver CSE received the Request. It does NOT indicate whether the Receiver CSE was able to take on responsibility for delivery of the data. In case the Originator CSE asked for the status of the requested Operation to be contained in the Result of the requested Operation (***rc*** not set to "nothing"), the Receiver CSE shall respond with a Success or Failure indicator. In case the Originator CSE asked for the status of the requested Operation and the URI of the created Resource to be contained in the Result of the Request, the Receiver CSE shall respond with a Success indicator including the URI of the created <delivery> resource in case it has taken on responsibility to deliver the data to the target CSE or with Failure indicator including an error indication otherwise.
Post-Processing at Originator	According to clause 10.1.1.1 with the following specific processing: The Originator CSE shall update the local <delivery> resource to reflect the new status of the delivery process (e.g., '{Receiver-CSE-ID} accepted delivery responsibility'). In case the Originator CSE got a Success indicator as a Response, it shall stop any further delivery attempts. In that case or if there was no indication of a need to provide a result of the operation, the Originator CSE may delete the content of the *'data'* attribute of the local <delivery> resource. In case the Originator CSE got a Failure indicator as a response, it may initiate further delivery attempts in line with CMDH policies and delivery parameters and depending on the reason for Failure. In case the Receiver CSE is the target CSE of the delivery, the Receiver CSE needs to execute on the forwarded request contained in the delivered data.
Exceptions	According to clause 10.1.1.1 with the following: • The Originator CSE is not authorized to request a delivery procedure on the Receiver CSE • The provided content of the <delivery> resource is not in line with the specified structure. • The provided content of the <delivery> resource represents a request for delivery that is not consistent (e.g., lifespan attribute already expired)

<delivery> CREATE	
	- The provided content of the <delivery> resource represents a request for delivery that cannot be met by the Receiver CSE within the limits of the provided delivery parameters and the provisioned CMDH policies on the Receiver CSE.

10.2.5.3 Retrieve <delivery>

This procedure shall be used for requesting a CSE to provide information on a previously created <delivery> resource which represents delivery of data to a target CSE.

Table 10.2.5.3-1: <delivery> RETRIEVE

<delivery> RETRIEVE	
Associated Reference Point	Mcc, Mcc’
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: void
Pre-Processing at Originator	According to clause 10.1.2 According to clause 10.1.1.1 with the following specific processing: Originator needs to retrieve information about a previously issued delivery
Processing at Receiver	According to clause 10.1.2 According to clause 10.1.1.1 with the following specific processing: The Receiver shall provide the content of the addressed <delivery> resource or the addressed attributes thereof.
Information on Response message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: attributes of the <delivery> resource as defined in clause 9.6.11
Post-Processing at Originator	According to clause 10.1.2.
Exceptions	According to clause 10.1.2 According to clause 10.1.1.1 with the following:; • The Originator CSE is not authorized to retrieve the <delivery> resource or the addressed parts of it • The addressed <delivery> resource does not exist

10.2.5.4 Update <delivery>

This procedure shall be used for requesting a CSE to update information on a previously created <delivery> resource which represents a pending delivery of data to a target CSE. The update may have impact on further processing of the delivery.

Table 10.2.5.4-1: <delivery> UPDATE

<delivery> UPDATE	
Associated Reference Point	Mcc, Mcc’
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: • URI of the <delivery> resource • Content of a <delivery> resource in line with the definition in clause 9.6.11 representing a valid request for delivery of data to a target CSE
Pre-Processing at Originator	According to clause 10.1.3 According to clause 10.1.1.1 with the following specific processing: Originator needs to modify information about a previously issued delivery that is still pending, i.e. it has not yet been forwarded to another CSE
Processing at Receiver	According to clause 10.1.3 wit According to clause 10.1.1.1 with the following specific processing: Receiver CSE checks if the requested changes to the delivery process can actually be accomplished If possible, the Receiver CSE modifies the previously established delivery process and changes the respective content of the <delivery> resource
Information on Response message	According to clause 10.1.3
Post-Processing at Originator	According to clause 10.1.3
Exceptions	According to clause 10.1.3 According to clause 10.1.1.1 with the following: • The Originator CSE is not authorized to modify the <delivery> resource or the addressed parts of it • The addressed <delivery> resource does not exist • The responsibility for the further processing of the delivery process represented by the addressed <delivery> process was already forwarded to another CSE

10.2.5.5 Delete <delivery>

This procedure shall be used for requesting a CSE to cancel a pending delivery of data to a target CSE or to delete the <delivery> resource of an already executed delivery.

Table 10.2.5.5-1: <delivery> DELETE

<delivery> DELETE	
Associated Reference Point	Mcc, Mcc’
Information on Request message	All parameters defined in table 8.1.2.1-1 apply
Pre-Processing at Originator	According to clause 10.1.4 with the following Originator needs to cancel a previously issued delivery that is still pending, i.e. it has not yet been forwarded to another CSE or Originator needs to remove the <delivery> resource representing an already executed delivery
Processing at Receiver	According to clause 10.1.4 • Receiver CSE checks if the corresponding delivery process is still pending. If so, it stops that delivery process • Receiver CSE removes the addressed <delivery> resource and stop the corresponding delivery process if it is still pending
Information on Response message	According to clause 10.1.4 with the following specific information: Successful Response messages indicate that the delivery process was stopped as requested
Post-Processing at Originator	According to clause 10.1.4
Exceptions	According to clause 10.1.4 with the following • The Originator CSE is not authorized to delete the <delivery> • The addressed <delivery> resource does not exist

10.2.6 Resource Discovery Procedure

10.2.6.1 Introduction

The resource discovery procedures allow discovering of resources residing on a CSE. The use of the *filterCriteria* parameter allows limiting the scope of the results.

Resource discovery shall be accomplished using the RETRIEVE method by an Originator which shall also include the root of where the discovery begins: e.g., <CSEBase>. The unfiltered result of the resource discovery procedure includes all the child resources under the root of where the discovery begins, which the Originator has a Discover access right on.

Filter criteria conditions may be provided as parameters to the Retrieve method. The filter criteria conditions describe the rules for resource discovery, e.g. resource types, creation time and matching string. The filter criteria can also contain the parameters for specifying the maximum size of the answer (upper limit), and/or sorting criteria for specifying in which order the searching result should be organized. Table 8.1.2-1 describes *filterCriteria* parameter.

A match shall happen when a resource matches all the configured filter criteria conditions and Originator has a Discover access right on the resource. A successful response contains a list for the matched resources addressable in any of the forms expressed in clause 9.3.1. However, if *Disrestype* parameter is specified in a discovery request, the hosting CSE shall choose the addressing form specified at *Disrestype* parameter.

The discovery results may be modified by the hosting CSE to restrict the scope of discoverable resources according to the Originator's access control policy or M2M service subscription.

The hosting CSE may also implement a configured upper limit on the size of the answer. In such a case when both the Originator and the hosting CSE have the upper limits, the smaller of the upper limit in the hosting CSE and the upper limit of the Originator shall apply.

10.2.6.2 Discovery procedure via Retrieve Operation

This procedure shall be used for the discovery of resources under <CSEBase> that match the provided *filterCriteria* attribute. The discovery result shall be returned to the Originator using a successful message.

[10.2.6.2.a] Editor's Note: It is FFS about discovery propagation mechanism, i.e. multi-hop discovery propagation and its policies.

Table 10.2.6.2-1: Discovery procedure via Retrieve Operation

<resource> RETRIEVE	
Associated Reference Point	Mca, Mcc, Mcc'
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: • URI of the discovery resource: e.g. <*CSEBase*> • Filter criteria for searching and expected returned result
Pre-Processing at Originator	According to clause 10.1.2 with the following: • ???
Processing at Receiver	According to clause 10.1.2 with the following specific processing: • Checks the validity of the Request (e.g., format of *filterCriteria*) • Checks if the request is in accordance with the M2M service subscription • May change the filter criteria according to local policies • Searches matched resources from the addressed resource hierarchy • Propagates the discovery request to other CSEs if it is permitted by discovery policies • Limits the discovery result according to access control privileges of the discovered resources • Limits the discovery result according to the upper limit on the size of the answer • Sorts the results according to the sorting criteria The hosting CSE shall read the values of all attributes belonging to the addressed resource structure and the references of all sub-resources and it shall build a representation of these. The hosting CSE shall use the appropriate addressing (see clause 9.3.1) for each element included in the list in accordance with the incoming request. If *filterCriteria* is provided in the request, the hosting CSE uses it identifying the resources whose attributes match the *filterCriteria*. The hosting CSE shall respond to the Originator with the appropriate list of discovered resources in the hosting CSE. If sorting criteria has been provided by the Originator, the list of discovered resources shall be sorted in that order. The hosting CSE may modify the *filterCriteria* including upper limit provided by the Originator or the discovery results based on the local policies. If the size of the result list is bigger than the upper limit or the scope of discoverable resources, according to the Originator's access control policy or service subscription has been reduced by the hosting CSE, the full list is not returned. Instead, an incomplete list is returned and an indication is added in the response for warning the device. The hosting CSE may propagate the discovery request to other CSEs according to resource discovery policies.
Information on Response message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: • Contains the URI list of discovered resources expressed in any of the methods depicted in clause 9.3.1. • Contains an incomplete list warning if the full list is not returned
Post-Processing at Originator	According to clause 10.1.2.
Exceptions	According to clause 10.1.2, with the following: • The requesting M2M AE or CSE is not registered • The request contains invalid parameters

10.2.7 Group Management Procedures

10.2.7.1 Introduction

This clause describes different procedures for managing membership verification, creation, retrieval, update and deletion of the information associated with a <group> resource as well as the bulk management of all group member resources by invoking the corresponding operations upon the virtual resource <fanOutPoint> of a <group> resource.

10.2.7.2 Create <group>

This procedure shall be used for creating a group resource.

Originator: The Originator shall request to Create a new group type resource to be named as <group> by using the CREATE operation. The request shall address <CSEBase> resource of a hosting CSE. The Request shall also provide list of member URI and may provide expirationTime attributes. The list of member URI means a list of URIs of the member resources corresponding to the memberType attribute provided in the Request. The originator may be an AE or a CSE.

Receiver: For the CREATE procedure, the Receiver shall:

- Check if the Originator has CREATE permissions on the <CSEBase> resource.
- Check the validity of the provided attributes.
- Validate that the resource type of every member conforms to the *memberType* attribute of the <group> resource, if the memberType attribute of the <group> resource is not 'mixed'. Set the *memberTypeValidated* attribute to TRUE upon successful validation.
- Upon successful validation of the provided attributes, create a new group resource including the <fanOutPoint> child-resource in the hosting CSE.
- Conditionally, in the case that the group resource contains temporarily unreachable sub-group resources as member resource, set the *memberTypeValidated* attribute of the <group> resource to FALSE.
- Respond to the Originator with the appropriate generic Response with the representation of the <group> resource if the *memberTypeValidated* attribute is FALSE, and the URI of the created <group> resource if the CREATE was successful.
- As soon as any unreachable resource becomes reachable, the *memberType* validation procedure shall be performed. If the *memberType* validation fails, the hosting CSE shall deal with the <group> resource according to the policy defined by the *consistencyStrategy* attribute of the <group> resource provided in the request. or by default if the attribute is not provided.

Table 10.2.7.2-1: <group> CREATE

Description	
Call Flow Type	CREATE
Pre-Conditions	None
Information on Request message	***op***: C ***fr***: Identifier of the AE or the CSE that initiates the Request ***to***: The URI of the <*CSEBase*> where the <group> resource is intended to be Created. ***cn***: The representation of the <group> resource for which the attributes are described in clause 9.6.13.
Local processing on Hosting CSE	Steps described for the Receiver of the CREATE Request as described above
Information on Response message	The representation of the <group> resource if the *memberTypeValidated* attribute is FALSE
Post-Conditions	None
Exceptions	No change from the generic procedure

10.2.7.3 Retrieve <group>

This procedure shall be used for retrieving <group> resource.

Originator: The Originator shall request to obtaining <group> resource information by using the RETRIEVE operation. The request shall address the specific <group> resource of a hosting CSE. The Originator may be an AE or a CSE.

Receiver: The Receiver shall check if the Originator has READ permission on the group resource. Upon successful validation, the hosting CSE shall respond to the Originator with the appropriate response and resource representation.

Table 10.2.7.3-1: <group> RETRIEVE

Description	
Call Flow Type	RETRIEVE
Pre-Conditions	None
Information on Request message	***op***: R ***fr***: Identifier of the AE or the CSE that initiates the Request ***to***: The URI of the <group> resource
Local processing on Hosting CSE	Same as the generic procedure
Information on Response message	Same as the generic procedure
Post-Conditions	None
Exceptions	No change from the generic procedure

10.2.7.4 Update <group>

This procedure shall be used for updating an existing <group> resource.

Originator: The Originator shall request to Update attributes of an existing <group> resource by using an UPDATE operation. The Request shall address the specific <group> resource of a CSE. The Originator may be an AE or a CSE.

Receiver: The UPDATE procedure shall be:

- Check if the Originator has WRITE permissions on the <group> resource.
- Check the validity of provided attributes.
- Validate that the resource type of every member conforms to the *memberType* attribute of the <group> resource, if the *memberType* attribute of the <group> resource is not 'mixed'. Set the *memberTypeValidated* attribute to TRUE upon successful validation.
- Upon successful validation of the provided attributes, update the group resource in the hosting CSE.
- Conditionally, in the case that the group resource contains temporarily unreachable sub-group resources as members resource set the *memberTypeValidated* attribute of the <group> resource to FALSE.
- Respond to the Originator with the appropriate generic response with the representation of the <group> resource if the *memberTypeValidated* attribute is FALSE, and the URI of the created <group> resource if the UPDATE is successful.
- As soon as any unreachable resource becomes reachable, the *memberType* validation procedure shall be performed. If the *memberType* validation fails, the hosting CSE shall deal with the <group> resource according to the policy defined by the *consistencyStrategy* attribute of the <group> resource provided in the request, or by default if the attribute is not provided.

Table 10.2.7.4-1: <group> UPDATE

Description	
Call Flow Type	UPDATE
Pre-Conditions	None
Information on Request message	***op***: U ***fr***: Identifier of the AE or the CSE that initiates the Request ***to***: The URI of the <group> resource
Local processing on Hosting CSE	Steps described for the Receiver of the UPDATE Request as described above
Information on Response message	The representation of the <group> resource if the *memberTypeValidated* attribute is FALSE
Post-Conditions	None
Exceptions	No change from the generic procedure

10.2.7.5 Delete <group>

This procedure shall be used for deleting an existing <group> resource.

Originator: The Originator shall request to delete an existing <group> type resource by using the DELETE operation. The request shall address the specific <group> resource of a hosting CSE. The Originator may be an AE or a CSE.

Receiver: The Receiver shall check if the Originator has DELETE permission on the <group> resource. Upon successful validation, the CSE shall remove the resource from its repository and shall respond to the Originator with the appropriate responses.

Table 10.2.7.5-1: <group> DELETE

Description	
Call Flow Type	DELETE
Pre-Conditions	None
Information on Request message	***op***: D ***fr***: Identifier of the AE or the CSE that initiates the Request ***to***: The URI of the <group> resource
Local processing on Hosting CSE	Same as the generic procedure
Information on Response message	Same as the generic procedure
Post-Conditions	None
Exceptions	No change from the generic procedure

10.2.7.6 Create <fanOutPoint>

This procedure shall be used for creating the content of all members resources belonging to an existing <group> resource.

Originator: The Originator shall request to create the content in all members resources belonging to an existing <group> resource by using a CREATE operation. The Request may address the virtual child resource <fanOutPoint> of the specific <group> resource of a group hosting CSE to create the same content under all <members> resources. The request may also address the URI that results from appending a relative URI to the <fanOutPoint> URI in order to create the same content (e.g. attribute or child resource) under the corresponding attributes or child resources represented by the relative URI with respect to all members resources. The Originator may be an AE or CSE.

Group Hosting CSE: For the CREATE procedure, the Group Hosting CSE shall:

- Check if the Originator has WRITE permission in the accessControlPolicy resource referenced by the members *AccessControlPolicyIDs* in the <group> resource. In the case members *membersAccessControlPolicyIDs* is not provided the access control policy defined for the <group> resource shall be used.
- Upon successful validation, obtain the URIs of all members resources from the attribute *membersList* of the addressed <group> resource.
- Generate fan out requests addressing the obtained URIs (appended with the relative URI if any) to the member hosting CSEs as indicated in Figure 10.2.2.6-1.The ***fr*** parameter in the request is set to ID of the Originator from the request from the original Originator.
- In the case that the members resources contain a sub-group resource, generate a unique group request identifier, include the group request identifier in all the requests to be fanned out and locally store the group request identifier.
- If the group hosting CSE determines that multiple members resources belong to one CSE according to the URIs of the members resources, it may converge the requests accordingly before sending out. This may be accomplished by the group hosting CSE creating a <group> resource on the members hosting CSE to collect all the members on that members hosting CSE.
- After receiving the responses from the members hosting CSEs, respond to the Originator with the aggregated results and the associated memberIDs.

Member Hosting CSEs: For the CREATE procedure, the Member Hosting CSE shall:

- Perform the corresponding CREATE procedure for the resource type in the request (clause 10.2 resource specific procedures).
- Check if the request has a group request identifier. Check if the group request identifier is contained in the requested identifiers stored locally. If match is found, ignore the current request and respond an error. If no match is found, locally store the group request identifier.
- Check if the original Originator has the CREATE permission on the addressed resource. Upon successful validation, perform the create procedures for the corresponding type of addressed resource as described in other sub-clauses of clause 10.2.
- Send the corresponding response to the Group Hosting CSE.

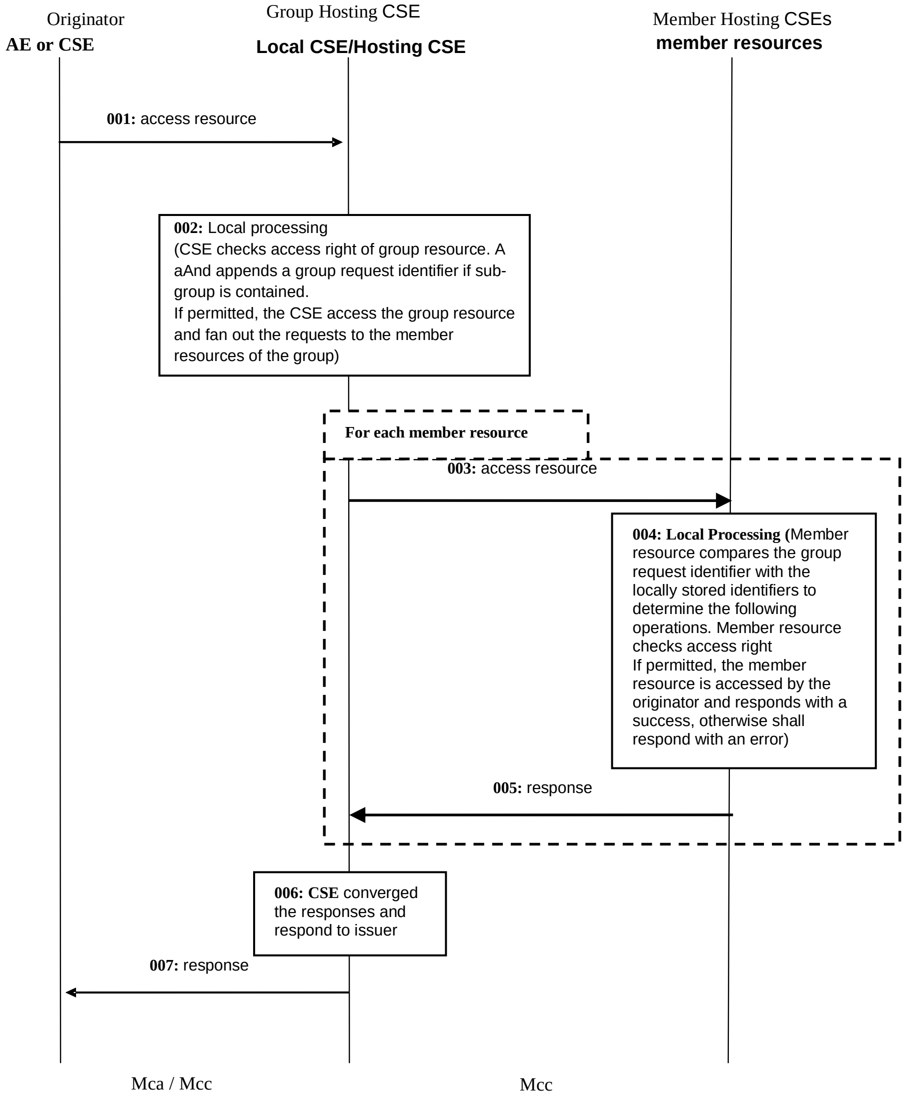

Originator
AE or CSE
Group Hosting CSE
Local CSE/Hosting CSE
Member Hosting CSEs
member resources
001: access resource
002: Local processing
(CSE checks access right of group resource. A aAnd appends a group request identifier if sub-group is contained.
If permitted, the CSE access the group resource and fan out the requests to the member resources of the group)
For each member resource
003: access resource
004: Local Processing (Member resource compares the group request identifier with the locally stored identifiers to determine the following operations. Member resource checks access right
If permitted, the member resource is accessed by the originator and responds with a success, otherwise shall respond with an error)
005: response
006: CSE converged the responses and respond to issuer
007: response
Mca / Mcc
Mcc

Figure 10.2.7.6-1: Group content management procedures

The procedures illustrated in figure 10.2.7.6-1 apply to clauses 10.2.7.6 to 10.2.7.9.

Table 10.2.7.6-1: <fanOutPoint> CREATE

CREATE: Description	
Pre-Conditions	None
Information on Request message	***op*****:** C ***fr*****:** Identifier of the AE or the CSE that initiates the Request ***to*****:** The URI of the <fanOutPoint> virtual resource ***cn*****:** The representation of the resource the Originator intends to create ***gid*****:** The group request identifier
Local processing on Hosting CSE	Fan out requests to each member hosting CSE, addressing the obtained URI from the attribute *membersList* of the group resource appended with relative URI if any Generate group request identifier and include the identifier in the fanned out requests in the case of sub groups
Information on Response message	Converged responses from members hosting CSEs
Post-Conditions	None
Exceptions	Same request with identical group request identifier received Originator does not have the access control privileges to access the <fanOutPoint> resource

10.2.7.7 Retrieve <fanOutPoint>

This procedure shall be used for retrieving the content of all member resources belonging to an existing <group> resource.

Originator: The Originator shall request to obtain the content or specific information (e.g. attributes) of all member resources belonging an existing <group> resource by using a RETRIEVE operation. The request may address the virtual child resource <fanOutPoint> of the specific <group> resource of a group hosting CSE for retrieving the content of all member resources. The request may also address the URI that results from appending a relative URI to the <fanOutPoint> URI in order to retrieve the corresponding attributes or child resources represented by the relative URI with respect to all members resources. The originator may be an AE or CSE.

Group Hosting CSE: For the RETRIEVE procedure, the Group Hosting CSE shall:

- Check if the Originator has READ permission in the accessControlPolicy resource referenced by the *membersAccessControlPolicyIDs* in the addressed <group> resource. In the case *membersAccessControlPolicyIDs* is not provided, the access control policy defined for the group resource shall be used.

- Upon successful validation, obtain the URIs of all members resources from the *membersList* attribute of the addressed <group> resource.

- Generate fan out requests addressing the obtained URIs (appended with the relative URI if any) to the members hosting CSEs as indicated in Figure 10.2.2.6-1.The ***fr*** parameter in the request is set to ID of the Originator from the request from the original Originator.

- In the case that the members resources contain a sub-group resource, generate a unique group request identifier, include the group request identifier in all the requests to be fanned out and locally store the group request identifier.

- If the group hosting CSE determines that multiple members resources belong to one CSE according to the URIs of the members resources, it may converge the requests accordingly before sending out. This may be accomplished by the group hosting CSE creating a <group> resource on the members hosting CSE to collect all the members on that members hosting CSE.

- After receiving the responses from the members hosting CSEs, respond to the originator with the aggregated results and the associated memberIDs.

Member Hosting CSEs: For the RETRIEVE procedure, the Member Hosting CSE shall:

- Perform the corresponding RETRIEVE procedure for the resource type in the request (clause 10.2 resource specific procedures).

- Check if the request has a group request identifier. Check if the group request identifier is contained in the requested identifier stored locally. If match is found, ignore the current request and respond an error. If no match is found, locally store the request identifier.

- Check if the original originator has the READ permission on the addressed resource. Upon successful validation, perform the retrieve procedures for the corresponding type of addressed resource as described in other sub-clauses of clause 10.2.

- Send the corresponding response to the group hosting CSE.

Table 10.2.7.7-1: <fanOutPoint> RETRIEVE

RETRIEVE: Description	
Pre-Conditions	None
Information on Request message	***op***: R ***fr***: Identifier of the AE or the CSE that initiates the Request ***to***: The URI of the <fanOutPoint> resource ***gid***: The group request identifier
Local processing on Hosting CSE	Fan out requests to each member hosting CSE, addressing the obtained URI from the attribute *membersList* of the group resource appended with relative URI if any Generate group request identifier and include the identifier in the fanned out requests in the case of sub groups
Information on Response message	Converged responses from member hosting CSEs
Post-Conditions	None
Exceptions	Same request with identical request identifier received Originator does not have the access control privileges to access the <fanOutPoint> resource

10.2.7.8 Update <fanOutPoint>

This procedure shall be used for updating the content of all member resources belonging to an existing <group> resource.

Originator: The Originator shall request to update the content of all member resources belonging to an existing <group> resource with the same new data by using a UPDATE operation. The request may address the virtual child resource <fanOutPoint> of the specific <group> resource of a group hosting CSE to update all <members> resources. The request may also address the URI that results from appending a relative URI to the <fanOutPoint> in order to update only the corresponding attributes or child resources represented by the relative URI with respect to all <members> resources. The originator may be an AE or CSE.

Group Hosting CSE: For the UPDATE procedure, the Group Hosting CSE shall:

- Check if the originator has WRITE permission in the <accessControlPolicy> resource referenced by the *membersAccessControlPolicyIDs* in the group resource. In the case members *membersAccessControlPolicyIDs* is not provided the access control policy defined for the group resource shall be used.

- Upon successful validation, obtain the URIs of all member resources from the attribute *membersList* of the addressed <group> resource.

- Generate fan out requests addressing the obtained URIs (appended with the relative URI if any) to the members hosting CSEs as indicated in figure 10.2.2.6-1.The ***fr*** parameter in the request is set to ID of the Originator from the request from the original Originator.

- In the case that the members resources contain a sub-group resource, generate a unique group request identifier, include it in all the requests to be fanned out and locally store the group request identifier.

- If the group hosting CSE determines that multiple members resources belong to one CSE according to the URIs of the member resources, it may converge the requests accordingly before sending out. This may be accomplished by the group hosting CSE creating a <group> resource on the member hosting CSE to collect all the members on that members hosting CSE.

- After receiving the responses from the member hosting CSEs, respond to the originator with the aggregated results and the associated memberIDs.

Member Hosting CSEs: For the UPDATE procedure, the Member Hosting CSE shall:

- Perform the corresponding UPDATE procedure for the resource type in the request (clause 10.2 resource specific procedures).
- Check if the request has a group request identifier. Check if the request identifier is contained in the requested identifier stored locally. If match is found, ignore the current request and respond an error. If no match is found, locally store the request identifier.
- Check if the original originator has the UPDATE permission on the addressed resource. Upon successful validation, perform the update procedures for the corresponding type of addressed resource as described in other sub-clauses of clause 10.2.
- Send the corresponding response to the group hosting CSE.

Table 10.2.7.8-1: <fanOutPoint> UPDATE

UPDATE: Description	
Pre-Conditions	None
Information on Request message	***op*:** U ***fr*:** Identifier of the AE or the CSE that initiates the Request ***to*:** The URI of the <group> resource ***cn*:** The representation of the resource the Originator intend to Update ***gid*:** The group request identifier
Local processing on Hosting CSE	Fan out requests to each members hosting CSE addressing the obtained URI from the attribute *membersList* of the group resource appended with relative URI if any Generate group request identifier and include the identifier in the fanned out requests in the case of sub groups
Information on Response message	Converged responses from member hosting CSEs
Post-Conditions	None
Exceptions	Same request with identical request identifier received Originator does not have the access control privileges to access the <fanOutPoint> resource

10.2.7.9 Delete <fanOutPoint>

This procedure shall be used for deleting the content of all members resources belonging to an existing <group> resource.

Originator: The Originator shall request to delete the content of all members resources belonging to an existing <group> resource by using a DELETE operation. The request may address the virtual child resource <fanOutPoint> of the specific <group> resource of a group hosting CSE to delete all members resources. The request may also address the URI that results from appending a relative URI to the <fanOutPoint> in order to delete only the corresponding attributes or child resources represented by the relative URI with respect to all member resources. The originator may be an AE or a CSE.

<Group> Hosting CSE: For the DELETE procedure, the <group> Hosting CSE shall:

- Check if the Originator has WRITE permission in the accessControlPolicy resource referenced by the *membersAccessControlPoliciIDs* in the <group> resource. In the case *membersAccessControlPolicyIDs* is not provided the access control policy defined for the group resource shall be used.
- Upon successful validation, obtain the URIs of all member resources from the attribute membersList of the addressed <group> resource.
- Generate fan out requests addressing the obtained URIs (appended with the relative URI if any) to the member hosting CSEs as indicated in figure 10.2.2.6-1. ***fr*** parameter in the request is set to ID of the Originator from the request from the original Originator.

- In the case that the members resources contain a sub-group resource, generate a unique group request identifier, include the group request identifier in all the requests to be fanned out and locally store the group request identifier.

- If the <group> hosting CSE determines that multiple members resources belong to one CSE according to the URIs of the members resources, it may converge the requests accordingly before sending out. This may be accomplished by the group hosting CSE creating a <group> resource on the member hosting CSE to collect all the members on that member hosting CSE.

- After receiving the responses from the members hosting CSEs, respond to the originator with the aggregated results and the associated memberIDs.

Members Hosting CSEs: For the DELETE procedure, the Members Hosting CSE shall:

- Perform the corresponding DELETE procedure for the resource type in the request (clause 10.2 resource specific procedures).

- Check if the request has a group request identifier. Check if the group request identifier is contained in the requested identifier stored locally. If match is found, ignore the current request and respond an error. If no match is found, locally store the group request identifier.

- Check if the original originator has the DELETE permission on the addressed resource. Upon successful validation, perform the delete procedures for the corresponding type of addressed resource as described in other sub-clauses of clause 10.2.

- Send the corresponding response to the Group Hosting CSE.

Table 10.2.7.9-1: <fanOutPoint> DELETE

DELETE: Description	
Pre-Conditions	None
Information on Request message	***op***: D ***fr***: Identifier of the AE or the CSE that initiates the Request ***to***: The URI of the <group> resource ***gid***: The group request identifier
Local processing on Hosting CSE	Fan out requests to each members hosting CSE addressing the obtained URI from the attribute *membersList* of the group resource appended with relative URI if any Generate group request identifier and include the identifier in the fanned out requests in the case of sub groups
Information on Response message	Converged responses from member hosting CSEs.
Post-Conditions	None
Exceptions	Same request with identical request identifier received Originator does not have the access control privilege to access the <fanOutPoint> resource

10.2.7.10 Subscribe and Un-Subscribe <fanOutPoint> of a group

This procedure shall be used for receiving information about modifications of all member resources belonging to an existing <group> resource.

Originator: shall request to create a subscription resource under all member resources belonging to an existing <group> resource by using a CREATE operation. The request shall address the child resource <fanOutPoint> of the specific <group> resource of a group hosting CSE appended with the ID of the <subscription> resource to be created to subscribe to the modifications of all member resources. The request shall include an empty *aggregationURI* attribute if the originator wants the group hosting CSE to aggregate the notifications. The request shall include the required information and may include the optional information as described in subscription management clause 10.2.11. The originator may be an AE or a CSE.

Group Hosting CSE: For the subscribe/un-subscribe procedure, the <Group> Hosting CSE shall:

Check if the originator has WRITE permission in the <accessControlPolicy> resource referenced by the *membersAccessControlPolicyIDs* in the group resource. In the case *membersAccessControlPolicyIDs* is not provided the access control policy defined for the group resource shall be used.

If the subscription resource in the request contains an *aggregationURI* attribute, assign an URI to the *aggregationURI* of the subscription resource. In case the aggregationURI attribute has a value, maintain the mapping of the generated *aggregationURI* and the former *aggregationURI*.

Upon successful validation, obtain the URIs of all member resources from the attribute *membersList* of the addressed <group> resource and fan out requests to the members hosting CSEs addressing the obtained URIs appended with the ID of the <subscription> resource to be created.

If the group hosting CSE determines that multiple members resources belong to one CSE according to the URIs of the member resources, it may converge the requests accordingly before sending out. This may be accomplished by the <group> hosting CSE creating a <group> resource on the members hosting CSE to collect all the members on that members hosting CSE.

After receiving the responses from the members hosting CSEs, respond to the originator with the aggregated results and the associated memberIDs.

Members Hosting CSEs: For the subscribe/un-subscribe procedure, the Members Hosting CSE shall treat the request received from the group hosting CSE as a normal SUBSCRIBE request on the addressed member resource as if it comes from the original originator. Therefore the members hosting CSE shall:

Check if the original originator has the READ permission on the members resource.

Upon successful validation, perform the subscribe procedures for the corresponding type of member resource as described in clause 10.2.12.

Send the corresponding response to the group hosting CSE.

Table 10.2.7.10-1: <fanOutPoint> Subscribe/Un-subscribe

Description	
Call flow type	CREATE
Pre-conditions	none
Information on Request message	***op*: C** ***fr*:** Identifier of the AE or CSE that initiates the request. ***to*:** The URI of the <fanOutPoint> resource appended with the ID of the <subscription> resource to be created. **gid:** The group request identifier
Local processing on Hosting CSE	Fan out requests to each member hosting CSE addressing the obtained URI from the attribute membersList of the group resource appended with the ID of the <subscription> resource to be created. Generate request identifier and include the identifier in the fanned out requests in the case of sub groups.
Information on Response message	Converged responses from member hosting CSEs.
Post-condition	None
Exceptions	Same request with identical request identifier received. Originator does not have the access control privilege to access the <fanOutPoint> resource.

10.2.7.11 Aggregate the notifications by group

This procedure is used for the group hosting CSE to aggregate the notifications from member hosting CSEs and forward the aggregated notification to the subscriber.

Members Hosting CSEs: Whenever the resource that is subscribed-to is modified in a way that matches the policies as is specified in clause 9.6.8, notification needs to be sent to the subscribe, the Members Hosting CSE shall:

If the subscription resource contains *aggregationURI*, notify the subscriber at the *aggregationURI* and include the notificationURI in the notification. Otherwise, notify at the URI represented by the attribute notificationURI.

Group Hosting CSE: For the notification procedure, the Group Hosting CSE shall:

On receiving the notifications from the member hosting CSEs at the *aggregationURI*, validate if the notification is sent from its member resource and contain a notificationURI attribute.

Upon successful validation, aggregate the notifications which have the same notificationURI address of a single subscriber. Send the aggregated notification to the subscriber according to the notificationURI in the notification. In the case the addressed group is the member of another group through which the subscription is created the notification shall be sent according to the mapping of the *aggregationURI* of the two <group> hosting CSEs.

Wait for the response. After receiving the response, split the response and respond to the members hosting CSEs separately.

The group hosting CSE may stop aggregating the notifications when the *expirationTime* of the corresponding subscription expires.

Subscriber: shall treat every notification extracted from the aggregated notification as a separate notification received from the subscribed resource and generate corresponding responses. The subscriber shall aggregate the responses to these notifications and send the aggregated response to the group hosting CSE.

10.2.8 mgmtObj Management Procedures

10.2.8.1 Introduction

This clause describes the management procedures over Mca and Mcc reference points. If external management technologies are used for management, different operations addressing a <mgmtObj> resource (or its attributes or child resources) shall be translated into existing management commands and procedures performed on the mapped external management object on the managed entity. The Receiver in the following procedures is IN-CSE.

10.2.8.2 Create <mgmtObj>

This procedure shall be used to create a specific <mgmtObj> resource in the hosting CSE to expose the corresponding management function of a managed entity (i.e. M2M Device/Gateway) over the Mca reference point. Depending on the data model being used, the created <mgmtObj> resource may be a partial or complete mapping from the external management object onto the managed entity. If such an external management object is missing from the managed entity, it shall be added to the managed entity. Further operations performed on the created <mgmtObj> resource shall be converted by the hosting CSE into a corresponding device management action performed on the mapped external management object on the managed entity using external management technologies (e.g. OMA-DM [i.5] or BBF TR-069 [i.4]).

Table 10.2.8.2-1: <mgmtObj> CREATE

<mgmtObj> CREATE	
Associated Reference Point	Mcc and Mca
Information on Request message	**op:** C **fr:** Identifier of the AE or the CSE that initiates the Request **to:** The URI of the *<node>* where the <mgmtObj> resource is intended to be Created **cn:** The representation of the <mgmtObj> resource for which the attributes are described in clause 9.6.15
Pre-Processing at Originator	The Originator shall be an IN-AE, or a CSE on a managed entity: • The Originator is a CSE: In this case, the CSE first collects the original external management object (the management tree structure or also the value of the tree nodes if needed) of the local device and transforms the data into the <mgmtObj> resource representation, then requests the hosting CSE to create the corresponding <mgmtObj> resource. • The Originator is an AE: In this case, the AE requests the hosting- CSE to add the corresponding external management object to the managed entity by creating an <mgmtObj> resource in the hosting CSE. NOTE 1: The IN-CSE can create the <mgmtObj> resource locally by itself. The details are out of scope. In this case, the hosting CSE first collects the original external management object on the managed entity via external management technology (e.g. OMA DM [i.5], BBF TR-069 [i.4] or LWM2M [i.6]), then transforms the object into the <mgmtObj> resource representation and create the <mgmtObj> resource locally in the IN-CSE. NOTE 2: The <mgmtObj> resource can be created in the hosting CSE by other offline provisioning means which are out of scope.
Processing at Receiver	For the CREATE operation, the Receiver shall: • Check if the Originator has the CREATE privilege on the addressed <node> resource. Then check the validity of the provided attributes. • Upon successful validation, create a new <mgmtObj> resource in the hosting CSE with the provided attributes. • If the Originator is an AE: Check if there is existing management session between the management server and the managed entity. If not, request the management server to establish a management session towards the managed entity. Send the management request to the managed entity or to the management server to add the corresponding external management object to the managed entity based on external management technology. • Maintain the mapping relationship between the created <mgmtObj> resource and the external management object on the managed entity. • Respond to the Originator with the appropriate responses based on the response from the external management technologies. It shall also provide in the response the URI of the created new resource.
Information on Response message	Error code if the new external management object is not created
Post-Processing at Originator	None
Exceptions	The creation of the external management object is not allowed The created external management object already exists Corresponding external management object cannot be added to the managed entity for some reason (e.g. not reachable, memory shortage)

10.2.8.3 Retrieve <mgmtObj>

This procedure shall be used to retrieve information from an existing <mgmtObj> resource.

Table 10.2.8.3-1: <mgmtObj> RETRIEVE

<mgmtObj> RETRIEVE	
Associated Reference Point	Mcc and Mca
Information on Request message	**op:** R **fr:** Identifier of the AE or the CSE that initiates the Request **to:** The URI of the <mgmtObj> resource
Pre-Processing at Originator	The Originator shall be an AE, or a CSE on the managed entity.
Processing at Receiver	For the RETRIEVE operation, the Receiver shall: • Check if the Originator has the READ privilege on the addressed <mgmtObj> resource. • Upon successful validation, retrieve the corresponding <mgmtObj> resource. • Respond to the Originator with the appropriate response. • If the Originator is an AE and if the requested information of the <mgmtObj> resource is not available, identify the corresponding external management object on the managed entity according to the mapping relationship that the IN-CSE maintains. Check if there is an existing management session between the management server and the managed entity. If not, request the management server to establish a management session towards the managed entity. Send the management request to get the corresponding external management object from the managed entity based on the external management technology, then return the result to the Originator based on the response from the external management technology.
Information on Response message	Error code if the new external management object can not be retrieved.
Post-Processing at Originator	None
Exceptions	Corresponding external management object data cannot be retrieved from the managed entity (e.g. external management object not found).

10.2.8.4 Update <mgmtObj>

This procedure shall be used to update information of an existing <mgmtObj> resource.

Table 10.2.8.4-1: <mgmtObj> UPDATE

<mgmtObj> UPDATE	
Associated Reference Point	Mcc and Mca
Information on Request message	**op:** U **fr:** Identifier of the AE or the CSE that initiates the Request **to:** The URI of the <mgmtObj> resource **cn:** The representation of the <mgmtObj> resource for which the attributes are described in clause 9.6.15
Pre-Processing at Originator	The Originator shall be an IN-AE, or a CSE on a managed entity.
Processing at Receiver	For the UPDATE operation, the Receiver shall: • Check if the Originator has the WRITE privilege on the address <mgmtObj> resource. Check the validity of provided attributes if any. • Upon successful validation, update the corresponding attribute(s) of the <mgmtObj> resource accordingly. • If the Originator is an IN-AE, identify the corresponding external management object on the managed entity according to the mapping relationship it maintains. Check if there is an existing management session between the management server and the managed entity. If not, request the management server to establish a management session towards the managed entity. Send the management request to update the corresponding external management object in the managed entity accordingly based on external management technology. • Respond to the Originator with the appropriate response based on the response from the external management technology.
Information on Response message	Error code if the external management object can not be updated
Post-Processing at Originator	None
Exceptions	Corresponding external management object cannot be updated to managed entity (e.g. not reachable, external management object not found)

10.2.8.5 Delete <mgmtObj>

This procedure shall be used to delete an existing <mgmtObj> resource. An IN-AE uses this procedure to remove the corresponding external management object (e.g. an obsolete software package) from the managed entity.

Table 10.2.8.5-1: <mgmtObj> DELETE

<mgmtObj> DELETE	
Associated Reference Point	Mcc and Mca
Information on Request message	**op:** D **fr:** Identifier of the IN-AE, or the CSE that initiates the Request **to:** The URI of the <mgmtObj> resource.
Pre-Processing at Originator	The Originator shall be an IN-AE or CSE on a managed entity. • The Originator is a CSE on the managed entity: In this case, the CSE issues the request to the hosting CSE to hide the corresponding management function from being exposed by the <mgmtObj> resource. • The Originator is an IN-AE: In this case, the IN-AE requests the hosting CSE to delete the <mgmtObj> resource from the hosting CSE and to remove the corresponding external management object from the managed entity. NOTE 1: The hosting IN-CSE can delete the <mgmtObj> resource locally by itself. This internal procedure is out of scope. NOTE 2: The <mgmtObj> resource can be deleted in the hosting CSE by offline provisioning means which are out of scope.
Processing at Receiver	For the DELETE operation, the Receiver shall: • If the Originator is an IN-AE, identify the corresponding external management object on the managed entity according to the mapping relationship IN-CSEmaintains. Check if there is an existing management session between the management server and the managed entity. If not, request the management server to establish a management session towards the managed entity. The IN-CSE sends management request to remove the corresponding external management object from the managed entity based on external management technology. • Respond to the Originator with the appropriate generic responses based on the response from the external management technology.
Information on Response message	Error code if the external management object can not be deleted.
Post-Processing at Originator	None
Exceptions	Corresponding external management object cannot be deleted from managed entity (e.g. not reachable, external management object not found).

10.2.8.6 Execute <mgmtObj>

This procedure shall be used to execute a management command on a managed entity through an existing <mgmtObj> resource on the hosting CSE.

Table 10.2.8.6-1: <mgmtObj> EXECUTE

<mgmtObj> DELETE	
Associated Reference Point	Mcc and Mca
Information on Request message	**op:** D **fr:** Identifier of the IN-AE, or the CSE that initiates the Request **to:** The URI of the <mgmtObj> resource.
Pre-Processing at Originator	The Originator shall be an IN-AE. The Originator shall request to execute a management command which is represented by an <mgmtObj> resource or its attribute by using an UPDATE operation. The request shall address the executable <mgmtObj> resource or its attribute and shall contain an empty body. After the execution request, the Originator shall request to retrieve the execution result or status from the executable <mgmtObj> resource or its attribute/child resource by using a RETRIEVE operation as specified in clause 10.2.7.3.
Processing at Receiver	For the EXECUTE operation , the Receiver shall: • Check if the Originator has the WRITE privilege on the addressed <mgmtObj> resource or its attribute. • Check if there is an existing management session between the management server and the managed entity. If not, request the management server to establish a management session towards the managed entity. Send the management request to execute the corresponding management command (e.g. "Exec" in OMA DM [i.5]) on the managed entity based on external management technology. • Respond to the Originator with the appropriate response based on the response from the external management technology. If available, the response shall contain execution results. • Retrieve the execution result or status from the executable <mgmtObj> resource or its attribute, perform the procedures as described in clause 10.2.7.3. • Upon receiving a management notification (e.g. OMA-DM [i.5] "Generic Alert" message or BBF TR-069 [i.4] "Inform" message) from a managed entity regarding the execution result or status, the Receiver shall send the management request to retrieve the execution result or status of the external management object information received from the managed entity and update the corresponding <mgmtObj> resource or its attribute.
Information on Response message	Error code if the external management technology procedure can not be executed
Post-Processing at Originator	None
Exceptions	Corresponding external management technology procedure cannot be executed in managed entity (e.g. not reachable, external management object not found)

10.2.9 External Management Operations through <mgmtCmd>

10.2.9.1 Introduction

This clause describes how RESTful management operations may be performed using <mgmtCmd> resources over the Mca and Mcc reference points. The <mgmtCmd> resource, together with its attributes or sub-resources, may be used in the process of translating between RESTful operations and management commands and procedures from existing management technologies (e.g. BBF TR-069 [i.4]). These procedures can then be performed on the remote entity, using the Management Adapter and the procedures described in the following clauses.

10.2.9.2 Create <mgmtCmd>

A CREATE request is used by an Originator to create a specific <mgmtCmd> resource in a hosting CSE.

The created <mgmtCmd> resource will be mapping a RESTful method to management commands and/or procedures which may be translated from existing management protocols (e.g. BBF TR-069 [i.4]). At run-time the hosting CSE can expose the translated commands, over the Mcc reference point, to the remote entities (i.e. ASN/MN-CSE).

Originator: The Originator shall request to create a new <mgmtCmd > resource to be named as "mgmtCmd" by using a CREATE operation. The request shall address <CSEBase> resource of the hosting CSE. The request may also provide the attributes of the <mgmtCmd> resource to be created as described in clause 9.6.15 such as "cmdType".

The Originator may be:

- An AE registered to the IN-CSE.
- The CSE on the managed entity: In this case, the CSE transforms supported management command into the <mgmtCmd> resource representation, then requests the hosting CSE to create the corresponding <mgmtCmd> resource.

NOTE 1: The hosting IN-CSE in the network domain may also create the <mgmtCmd> resource locally by itself. The details are out of scope. Then an AE can discover the created <mgmtCmd> and manipulate it.

NOTE 2: The <mgmtCmd> resource could also be created in the hosting CSE by other offline provisioning means which are out of scope.

Receiver: The Receiver shall check if the Originator has the CREATE permission on the addressed <CSEBase> resource (or the parent <mgmtCmd> resource in the case of child resource creation). The hosting CSE shall also check the validity of provided attributes. Upon successful validation, a new resource with name "mgmtCmd" shall be created in the hosting CSE with the provided attributes. The hosting CSE shall maintain the mapping between the created <mgmtCmd> resource and the corresponding nonRESTful commands represented by the "*cmdType*" attribute of <mgmtCmd> resource. The hosting CSE shall respond to the Originator with the appropriate generic response code. It shall also provide in the Response the URL of the created new resource.

The hosting CSE shall be an IN-CSE.

Table 10.2.9.2-1: <mgmtCmd> CREATE

Description	
Call Flow Type	CREATE
Pre-Conditions	The CSE on the originating node shall first collect local management command.
Information on Request message	***op***: C ***fr***: Originator AE-ID or CSE-ID ***to***: Receiver CSE-ID ***cn***: Name of <mgmtCmd> resource. The mandatory and/or optional attributes defined in clause 9.6.15, as needed
Local processing on Hosting CSE	No change from the basic procedures
Information on Response message	***fr***: Receiver CSE-ID ***to***: Originator AE-ID or CSE-ID ***cn***: URI of created <mgmtCmd> resource
Post-Conditions	No change from the basic procedures
Exceptions	If the creation is not allowed, the *<mgmtCmd>* resource shall not be created, and a proper error code shall be returned to the originator in the Response message

10.2.9.3 Retrieve <mgmtCmd>

A RETRIEVE Request is used by an Originator to retrieve all or part information from an existing <mgmtCmd> resource on a hosting CSE. Alternatively, the Originator can request to retrieve only a specific attribute or part of an attribute.

Originator: The Originator shall request to retrieve all or part of the information from an existing <mgmtCmd> resource by using a RETRIEVE operation. The request shall address a specific "<mgmtCmd>" resource to retrieve all attributes and the references to the sub-resources of the <mgmtCmd> resource. Alternatively, the request shall address the individual attributes of the specific "<mgmtCmd>" resource to retrieve the corresponding attribute value.

[10.3.9.3.a] Editor's Note: We need to agree on a terminology on how to refer to a specific instance of a resource type.

The Originator may be:

- An AE.
- A CSE.

Receiver: The Receiver shall check if the Originator has the READ permission on the addressed <mgmtCmd> resource. Upon successful validation, the hosting CSE shall retrieve the corresponding attributes the <mgmtCmd> resource and the references to the sub-resources from its repository and shall respond to the Originator with the appropriate generic responses.

The hosting CSE shall be an IN-CSE.

Table 10.2.9.3-1: <mgmtCmd> RETRIEVE

Description	
Call Flow Type	RETRIEVE
Pre-Conditions	Originator needs to create a resource
Information on Request message	***op***: R ***fr***: Originator AE-ID or CSE-ID ***to***: Receiver CSE-ID ***cn***: Name of <mgmtCmd> resource. The mandatory and/or optional attributes defined in clause 9.6.15, as needed
Local processing on Hosting CSE	No change from the basic procedures
Information on Response message	No change from the basic procedures
Post-Conditions	No change from the basic procedures
Exceptions	If the retrieval is not allowed, or the specific <mgmtCmd> resource does not exist in the IN-CSE, a proper error code shall be returned to the originator

10.2.9.4 Update <mgmtCmd>

An UPDATE request is used by an Originator to update all or part information of an existing <mgmtCmd> resource on a hosting CSE with new attributes. Alternatively, the Originator can request to update only a specific attribute or part of an attribute.

Originator: The Originator shall request to update all or partial information of an existing <mgmtCmd> resource by using a UPDATE operation. The request shall address a specific "<mgmtCmd>" resource of a CSE to update all attributes of the <mgmtCmd> resource with the provided new value. Alternatively, the request shall address the individual attribute of the specific "<mgmtCmd>" resource to update the corresponding attribute with the provided new value.

The Originator may be:

- An AE.
- A CSE.

Receiver: The Receiver shall check if the Originator has the WRITE permission on the addressed <mgmtCmd> resource. The hosting CSE shall also check the validity of provided attributes if any. Upon successful validation, the hosting CSE shall overwrite the corresponding attributes of the <mgmtCmd> resource with the provided new data. The CSE shall respond to the originator with the appropriate generic responses.

The hosting CSE shall be an IN-CSE.

Table 10.2.9.4-1: <mgmtCmd> UPDATE

Description	
Call Flow Type	UPDATE
Pre-Conditions	Originator needs to retrieve a resource
Information on Request message	***op***: U ***fr***: Originator AE-ID or CSE-ID ***to***: Receiver CSE-ID ***cn***: Name of <mgmtCmd> resource. The mandatory and/or optional attributes defined in clause 9.6.15, as needed
Local processing on Hosting CSE	No change from the basic procedures
Information on Response message	No change from the basic procedures
Post-Conditions	No change from the basic procedures
Exceptions	If the update is not allowed or the specific *<mgmtCmd>* resource (or its sub-resource) does not exist in the IN-CSE, a proper error code shall be returned to the Originator

10.2.9.5 Delete <mgmtCmd>

A DELETE request is used by the Originator to delete an existing <mgmtCmd> resource on a hosting CSE. An AE may also use this procedure to cancel all initiated <execInstance> of an <mgmtCmd> if applicable.

Originator: The Originator shall request to delete an existing <mgmtCmd> resource by using a DELETE operation. The request shall address the specific "<mgmtCmd>" resource on the hosting CSE.

The Originator may be:

- The CSE on the manageable entity: In this case, the CSE issues the request to the hosting CSE to hide the corresponding management command from being exposed by the <mgmtCmd> resource.
- An AE: In this case, the AE requests the hosting CSE to delete the <mgmtCmd> resource from the hosting CSE and cancel all initiated <execInstance> of an <mgmtCmd> if applicable.

NOTE 1: The hosting CSE in the network domain could also delete an <mgmtCmd> resource locally by itself. This internal procedure is out of scope.

NOTE 2: The <mgmtCmd> resource could also be deleted in the hosting CSE by other offline provisioning means which are out of scope.

Receiver: The Receiver shall check if the Originator has the DELETE permission on the addressed <mgmtCmd> resource. Upon successful validation, the hosting CSE shall remove the resource from its repository. If the Originator is an AE and there is any initiated <execInstance> under the <mgmtCmd> that can be cancelled by a corresponding management command. The hosting CSE shall also issue the management command to the remote entity to cancel those initiated <execInstance> based on existing management protocol (i.e. BBF TR-069 [i.4]). Then the CSE shall respond to the originator with the appropriate generic responses.

The hosting CSE shall be an IN-CSE.

Table 10.2.9.5-1: <mgmtCmd> DELETE by ASN-CSE or MN-CSE

Description	
Call Flow Type	DELETE
Pre-Conditions	Before issuing a DELETE request to the IN-CSE, the originating CSE may perform cancelling of the corresponding management command locally.
Information on Request message	***op***: D ***fr***: Originator AE-ID or CSE-ID. ***to***: Receiver CSE-ID. ***cn***: Name of <mgmtCmd> resource. The mandatory and/or optional attributes defined in clause 9.6.15, as needed
Local processing on Hosting CSE	The <mgmtCmd> resource shall be deleted from the repository of the IN-CSE
Information on Response message	No change from the basic procedures
Post-Conditions	No change from the basic procedures
Exceptions	If the deletion is not allowed or the specific <mgmtCmd> resource does not exist in the IN-CSE, the deletion shall be failed and a proper error code shall be returned to the Originator

Table 10.2.9.5-2: <mgmtCmd> DELETE by an AE

Description	
Call Flow Type	DELETE
Pre-Conditions	Originator needs to delete the resource.
Information on Request message	***op***: D ***fr***: Originator AE-ID or CSE-ID ***to***: Receiver CSE-ID ***cn***: Name of <mgmtCmd> resource. The mandatory and/or optional attributes defined in clause 9.6.15, as needed
Local processing on Hosting CSE	If there is any initiated <execInstance> under <mgmtCmd> and it is cancellable, IN-CSE shall cancel those initiated <execInstance> from the remote entity using corresponding management procedures in existing management protocol (i.e. CancelTransfer RPC in BBF TR-069 [i.4]) The <mgmtCmd> resource shall be deleted from the repository of the IN-CSE
Information on Response message	No change from the basic procedures
Post-Conditions	No change from the basic procedures
Exceptions	If the deletion is not allowed or the specific <mgmtCmd> resource does not exist, there is no local processing in the IN-CSE and a proper error code shall be returned to AE If the corresponding initiated commands cannot be deleted from remote entity due to some reason (e.g. not found) a response with the proper indication shall be returned to AE

10.2.9.6 Execute <mgmtCmd>

The Execute procedure is used by an originator for execution of a specific management command on a remote entity, through an existing <mgmtCmd> resource on the hosting CSE.

Originator: The Originator shall request to execute a specific management command which is represented by an existing <mgmtCmd> resource by using an UPDATE operation. The UPDATE request shall address the execEnable attribute of the <mgmtCmd> resource without any payload. Alternatively, the UPDATE request shall address the URI provided as the value of the *execEnable* attribute of the <mgmtCmd> resource.

After issuing the execution request, the Originator may request to retrieve the execution result or status from an <execInstance> sub-resource of the <mgmtCmd>by using a RETRIEVE method as described in clause 10.2.7.2.

The Originator shall be an AE.

Receiver: The Receiver shall check if the Originator has the WRITE permission on the addressed <mgmtCmd> resource. Upon successful validation, the hosting CSE shall perform command conversion and mapping, and send the converted management command over ***mId*** reference point to execute the corresponding management command with the provided arguments on the remote entity based on existing device management protocol (i.e. BBF TR-069 [i.4]).

Then the hosting CSE shall create a corresponding <execInstance> resource under <mgmtCmd> for this command execution. And the hosting CSE shall respond to the originator with the appropriate generic responses. It shall also provide in the response the URL of the created <execInstance> resource.

Upon receiving from the remote entity a management notification (i.e. BBF TR-069 [i.4] "Inform" message) regarding the execution result or status, the hosting CSE may update the corresponding <execInstance> sub-resource locally.

The hosting CSE shall be an IN-CSE.

[10.2.9.6.a] Editor's note: The definition of Execute is to be for FFS in clause 10.1.

Table 10.2.9.6-1: <mgmtCmd> EXECUTE

Description	
Call Flow Type	UPDATE
Pre-Conditions	Originator needs to retrieve a resource.
Information on Request message	***op*****:** U. ***fr*****:** Originator AE-ID or CSE-ID. ***to*****:** Receiver CSE-ID. ***cn*****:** <mgmtCmd>/execEnbale The UPDATE request shall address to <mgmtCmd>/execEnable without any payload. The mandatory and/or optional attributes defined in clause 9.6.15, as needed.
Local processing on Hosting CSE	(a) If the execution is allowed, hosting CSE will do command conversion and mapping. (b) The hosting CSE shall trigger existing device management procedures (i.e. BBF TR-069 [i.4]) to execute the corresponding management command on the remote entity. (c) The hosting CSE shall perform local processing: if Step b is successful, the IN-CSE shall create an <execInstance> resource under the triggered <mgmtCmd> to maintain status and results for this execution.
Information on Response message	No change from the basic procedures.
Post-Conditions	The following processing on the hosting CSE is dependent on the type of the command and execution status and may occur after the Response message has been sent. (a) After the command execution is finished, the remote entity sends response including execution results to the hosting CSE, who will store the execution results in corresponding <execInstance> resource. This step may occur any time after the Response message has been sent and might occur after step b below. (b) The AE may use normal RETRIEVE procedure to retrieve the execution results or status of an <execInstance>. (c) After receiving the RETRIEVE request from the AE, the hosting CSE can retrieve the execution status or results on the remote entity using existing management protocol. If step d. occurs before step e, step f. may not be needed. (d) A response shall be returned to the AE.
Exceptions	If the execution is not allowed or the specified *<mgmtCmd>* resource or its attribute/sub-resource does not exist in the IN-CSE, no further processing is required in the hosting CSE and a proper error code shall be returned to AE in the Message response. If the corresponding management command cannot be executed in remote entity due to some reason, step 004 <??> shall be skipped and a proper error code shall be returned to AE in step 005 <??>.

[10.2.9.6.b] Editor's Note: Reference to setps 004 and 005 needed.

10.2.9.7 Cancel <execInstance>

The Cancel procedure is used by an originating AE to disable/stop/cancel an initiated management command execution on the remote entity, through an existing <execInstance> resource on the hosting CSE.

Originator: The Originator shall request to disable/stop/cancel an initiated management command execution which is represented by an existing <execInstance> resource, by using an UPDATE operation. The UPDATE request shall address the *execDisable* attribute without any payload.

The Originator shall be an AE.

Receiver: The Receiver shall check if the Originator has the WRITE permission on the addressed <mgmtCmd> resource. Upon successful validation, the hosting CSE shall perform command conversion and mapping, then use existing management protocol (i.e. BBF TR-069 [i.4]) to cancel the corresponding management command execution initiated on the remote entity, and the hosting CSE shall respond to the originator with the appropriate responses.

The hosting CSE shall be an IN-CSE.

Table 10.2.9.7-1: <execInstance> CANCEL

Description	
Call Flow Type	UPDATE
Pre-Conditions	Originator needs to disable/stop/cancel an initiated management command execution on the remote entity
Information on Request message	***op***: U ***fr***: Originator AE-ID or CSE-ID ***to***: Receiver CSE-ID ***cn***: <mgmtCmd>/execInstance. The mandatory and/or optional attributes defined in clause 9.6.15, as needed
Local processing on Hosting CSE	If the cancellation is allowed, IN-CSE will do command conversion and mapping The IN-CSE shall cancel the <execInstance> from the remote entity using corresponding management procedures in existing management protocol (i.e. CancelTransfer RPC in BBF TR-069 [i.4]) The IN-CSE shall delete the corresponding <execInstance> resource
Information on Response message	No change from the basic procedures
Post-Conditions	No change from the basic procedures
Exceptions	If the execution is not allowed or the specified <execInstance> resource does not exist in the IN-CSE or the <execInstance> is finished, the local processing on hosting CSE shall be skipped and a proper error code shall be returned to AE in the Response message If the corresponding management command cannot be executed in remote entity due to some reason, the IN-CSE shall still delete the corresponding <execInstance> resource and return a Response message

[10.2.9.7.a] Editor's Note: The Cancel flow shall be updated based on the definition of EXEC procedure.

10.2.9.8 Retrieve <execInstance>

A RETRIEVE request is used by an Originator to retrieve an existing <execInstance> resource, including individual attributes, from a hosting CSE. Alternatively, the Originator can request to retrieve only a specific attribute or part of an attribute.

Originator: The Originator shall request to retrieve all or part of information from an existing <execInstance> resource by using a RETRIEVE operation. The Request shall address a specific "<execInstance>" resource of a hosting CSE to retrieve all attributes of the <execInstance> resource. Or the request shall address the individual attribute of the specific "<execInstance>" resource to retrieve the corresponding attribute value.

The Originator shall be an AE.

Receiver: The Receiver shall check if the Originator has the READ permission on the addressed <mgmtCmd> resource. Upon successful validation, the CSE shall retrieve the corresponding attributes of the <execInstance> resource and the references to the sub-resources from its repository and shall respond to the originator with the appropriate generic responses.

The hosting CSE may be an IN-CSE.

Table 10.2.9.8-1: <execInstance> RETRIEVE

Description	
Call Flow Type	RETRIEVE
Pre-Conditions	Originator needs to create a resource.
Information on Request message	***op*: R** ***fr*:** Originator AE-ID or CSE-ID ***to*:** Receiver CSE-ID ***cn*:** <mgmtCmd> resource. The mandatory and/or optional attributes defined in clause 9.6.15, as needed
Local processing on Hosting CSE	If the retrieval is allowed, the IN-CSE can retrieve the execution status or results on the remote entity using existing management protocol (i.e. BBF TR-069 [i.4]) If the retrieval is allowed, the addressed attributes of the <execInstance> resource shall be retrieved from the repository of the IN-CSE
Information on Response message	No change from the basic procedures
Post-Conditions	No change from the basic procedures
Exceptions	If the retrieval is not allowed or the specific <execInstance> resource does not exist in the IN-CSE, there is no local processing on the hosting CSE and a proper error code shall be returned to AE in the Response Message

10.2.9.9 Delete <execInstance>

The DELETE request is used by an originating AE to delete an existing <execInstance> resource on a hosting CSE.

Originator: The Originator shall request to delete an existing <execInstance> resource by using a DELETE operation. The request shall address the specific "<execInstance>" resource of an <mgmtCmd> on the hosting CSE.

The originator shall be an AE.

NOTE 1: The hosting CSE in the network domain could also delete an <execInstance> resource locally by itself. This internal procedure is out of scope.

NOTE 2: The <execInstance> resource could also be deleted in the hosting CSE by other offline provisioning means which are out of scope.

Receiver: The Receiver shall check if the Originator has the DELETE permission on the addressed <execInstance> resource. Upon successful validation, the hosting CSE shall remove the resource from its repository. If the <execInstance> is not finished on the remote entity and it is cancellable, the hosting CSE shall also use existing management protocols (i.e. BBF TR-069 [i.4] CancelTransfer RPC) to cancel the corresponding management currently initiated at the remote entity. Then the CSE shall respond to the originator with the appropriate generic responses.

The hosting CSE shall be an IN-CSE.

Table 10.2.9.9-1: <execInstance> DELETE

Description	
Call Flow Type	DELETE
Pre-Conditions	Originator needs to delete the resource
Information on Request message	***op***: D ***fr***: Originator AE-ID or CSE-ID ***to***: Receiver CSE-ID ***cn***: Name of <mgmtCmd> resource. The mandatory and/or optional attributes defined in clause 9.6.15, as needed
Local processing on Hosting CSE	If the <execInstance> is not finished yet and it is cancellable, the IN-CSE shall cancel the <execInstance> from the remote entity using corresponding management procedures in existing management protocol (i.e. CancelTransfer RPC in BBF TR-069 [i.4]) The <execInstance> resource shall be deleted from the repository of the IN-CSE. If the corresponding initiated commands cannot be deleted from remote entity due to some reason (e.g. not found), the <execInstance> resource shall be still deleted be performed
Information on Response message	No change from the basic procedures
Post-Conditions	No change from the basic procedures
Exceptions	If the deletion is not allowed or the specific <execInstance> resource does not exist in the IN-CSE, there is no processing on the hosting CSE and a proper error code shall be returned to AE in the Response message If the <execInstance> is already complete or it is not cancellable, there is no processing on the hosting CSE

10.2.10 Location Management Procedures

10.2.10.1 Procedure related to <locationPolicy> resource

This clause introduces the procedures for obtaining and managing a target M2M Node's location information, which are associated with the <locationPolicy> resource that contains the method for obtaining and managing location information.

10.2.10.1.1 Create <locationPolicy>

This procedure shall be used for creating a <locationPolicy> resource.

Originator: The Originator shall request to CREATE a <locationPolicy> resource including the relevant attributes and the address <CSEBase> resource of a hosting CSE. Minimally, the Request shall provide the mandatory attributes defined in the table 9.6.10-1. The Originator in this case is an AE or CSE.

Receiver: For the CREATE procedure, the Receiver (hosting CSE) shall:

- Check whether the Originator is authorized to request the procedure.
- Check whether the provided attributes of the <locationPolicy> resource represent a valid Request.
- Upon successful validation of the above procedures, the hosting CSE creates the <locationPolicy> resource and automatically creates <container> resource where the actual location information is/are stored and the resources shall contain cross-reference between the both resources: *locationContainerID* attribute for <locationPolicy> resource and *locationID* attribute for <container> resource.
- Check the defined *locationSource* attribute to determine which method is used.. The *locationSource* attribute shall be set based on the capabilities of a target M2M Node, the required location accuracy of the Originator and the Underlying Network in which a target M2M Node resides:
  - For the Network-based case, the hosting CSE shall transform the Request from the Originator into Location Server request following the attributes (e.g. *locationTargetID*, *locationServer*) defined in the <locationPolicy> resource. Additionally, the hosting CSE shall also provide default values for other

parameters (e.g. required quality of position) in the Location Server request [i.7] according to local policies. The request towards the Location Server crosses over the Mcn reference point. Then the Location Server in the Underlying Network performs positioning procedures, and returns the results over the Mcn reference point.

- The specific mechanism used to communicate with the network Location Server depends on the capabilities of the Underlying Network and other factors. For example, it could be either the OMA Mobile Location Protocol [i.7] or OMA RESTful NetAPI for Terminal Location [i.8].

NOTE: The details of the mechanisms are addressed in the oneM2M Core Protocol Specification [i.2].

- For the Device-based case, this case is applicable if the Originator is ASN-AE and the ASN has location determination capabilities (e.g. GPS). The hosting CSE is capable of performing positioning procedure using the module or technologies. For example, if the ASN has a GPS module itself, the ASN-CSE obtains the location information of Node from the GPS module through internal interfaces (e.g. System call or JNI [i.21]). The detail procedure is out-of-scope.

- For the Sharing-based case, this case shall be applicable if the Originator is an ADN-AE and the hosting CSE is MN-CSE and the ADN is a resource constrained node, no location determination capabilities (e.g. GPS) and Network-based positioning capabilities. Also according to the required location accuracy of the AE, the Originator may choose this case.

  When the hosting CSE receives the CREATE request and if the hosting CSE can find the closest Node that is registered with the hosting CSE and has location information from the Originator in the M2M Area Network, the location information of the closest Node shall be stored as the location information of the Originator, or if the hosting CSE cannot find any closest Node or has no topology information, the location information of the Node of the hosting CSE (MN) shall be stored as the location information of the Originator. The closest Node can be determined by the minimum hop based on the topology information stored in the <node> resource.

Table 10.2.10.1.1-1: <locationPolicy> CREATE

CREATE: Description	
Pre-Conditions	None
Information on Request message	***op*: C** ***fr***: Identifier of the AE or the CSE that initiates the Request ***to***: the URI of the <CSEBase> resource ***cn***: The representation of the <locationPolicy> resource described in clause 9.6.10
Local processing on Hosting CSE	Detail procedure for the hosting CSE of the CREATE request described above
Information on Response message	The representation of the created <locationPolicy> resource
Post-Conditions	Detail steps for the hosting CSE after the CREATE request are described above
Exceptions	No change from the generic procedure

10.2.10.1.2 Retrieve <locationPolicy>

This procedure shall be used for retrieving an existing <locationPolicy> resource.

Originator: The Originator shall request to obtain <locationPolicy> resource information by using RETRIEVE operation. The Originator is either an AE or a CSE.

Receiver: The Receiver shall check if the Originator has RETRIEVE permission on the <locationPolicy> resource. Upon successful validation, the hosting CSE shall respond to the Originator with the appropriate responses.

Table 10.2.10.1.2-1: <locationPolicy> RETRIEVE

RETRIEVE: Description	
Pre-Conditions	None
Information on Request message	***op***: R ***fr***: Identifier of the AE or the CSE that initiates the Request ***to***: the URI of the target <locationPolicy> resource
Local processing on Hosting CSE	No change from the generic procedure
Information on Response message	No change from the generic procedure
Post-Conditions	None
Exceptions	No change from the generic procedure

10.2.10.1.3 Update <locationPolicy>

This procedure shall be used for updating an existing <locationPolicy> resource.

Originator: The Originator shall request to update attributes of an existing <locationPolicy> resource by using an UPDATE operation. The request shall address the specific <locationPolicy> resource of a CSE. The Originator may be either an AE or a CSE.

Receiver: The Receiver of an UPDATE request shall check whether the Originator is authorized to request the operation. The receiver shall further check whether the provided attributes of the <locationPolicy> resource represent a valid request for updating <locationPolicy> resource. The updatable attributes are (excluding common attributes):

- *locationUpdatePeriod*: This value is updated to change the period for updating location information.

Table 10.2.10.1.3-1: <locationPolicy> UPDATE

UPDATE: Description	
Pre-Conditions	None
Information on Request message	***op***: U ***fr***: Identifier of the AE or the CSE that initiates the Request ***to***: the URI of the target <locationPolicy resource ***cn***: The attributes which are to be updated.
Local processing on Hosting CSE	No change from the generic procedure
Information on Response message	No change from the generic procedure
Post-Conditions	None
Exceptions	No change from the generic procedure

10.2.10.1.4 Delete <locationPolicy>

This procedure shall be used for deleting an existing <locationPolicy> resource.

Originator: The Originator shall request to delete an existing <locationPolicy> resource by using the DELETE operation. The Originator may be either an AE or a CSE.

Receiver: The Receiver shall check if the Originator has DELETE permission on the <locationPolicy> resource. Upon successful validation, the CSE shall remove the resource from its repository and shall respond to the Originator with appropriate responses.

Once the <locationPolicy> resource is deleted, the Receiver shall delete the associated resources (e.g. <container>, <contentInstance> resources). If the locationSource attribute and the locationUpdatePeriod attribute of the <locationPolicy> resource has been set with appropriate value, the Receiver shall tear down the session. The specific mechanism used to tear down the session depends on the support of the Underlying Network and other factors.

Table 10.2.10.1.4-1: <locationPolicy> DELETE

DELETE: Description	
Pre-Conditions	When the locationSource of the created <locationPolicy> resource is "sharing-based" and the AE disconnects from the registered MN-CSE
Information on Request message	***op***: D ***fr***: Identifier of the AE or the CSE that initiates the Request ***to***: the URI of the target <locationPolicy resource
Local processing on Hosting CSE	No change from the generic procedure
Information on Response message	No change from the generic procedure
Post-Conditions	None
Exceptions	No change from the generic procedure

10.2.10.2 Procedure when the <container> and <contentInstance> resource contain location information

Since the actual location information of a target M2M Node shall be stored in the <contentInstance> resource as per the configuration described in the associated <locationPolicy> resource, this clause introduces the procedures related to the <contentInstance> and <container> resource.

10.2.10.2.1 Procedure for <container> resource that stores the location information

This procedure is mainly triggered by the creation of <locationPolicy> resource. Based on the defined attributes related to the <container> resource such as '*locationContainerID'* and '*locationContainerName'*, the hosting CSE will create <container> resource to store the location information in its child resource, <contentInstance> resource after the CSE obtains the actual location information of a target M2M Node. Since the <container> is unnamed resource type, the name of the created <container> resource shall be determined by the '*locationContainerID* if it is available. After the creation of the <container> resource, the actual location of the resource shall be stored in the '*locationContainerID'*.

10.2.10.2.2 Procedure for <contentInstance> resource that stores location information

After the <container> resource that stores the location information is created, each instance of location information will be stored in the different <contentInstance> resources. In order to store the location information in the <contentInstance> resource, the hosting CSE firstly checks the defined '*locationUpdatePeriod'* attribute. If a valid period value is set for this attribute, the hosting CSE performs the positioning procedures as defined period value, locationUpdatePeriod, in the associated <locationPolicy> resource and stores the results (e.g. position fix and uncertainty) in the <contentInstanace> resource under the created location resource. However, if no value (e.g. null) is set, the positioning procedure is performed when the created <container> is retrieved and the result will be stored in the <contentInstance> resource.

10.2.11 <subscription> Resource Procedures

10.2.11.1 Introduction

An Originator can create a <subscription> resource on a subscribed-to resource hosting CSE to be notified when the resource is modified. After successful <subscription> resource creation, the hosting CSE shall notify the Originator of a subscribed-to resource modification that meets conditions configured in the <subscription> resource.

A subscription shall be represented by a <subscription> resource (see clause 9.6.8). This allows manipulation of the subscription in a resource oriented manner, e.g. the conditions of a subscription may be modified by modifying a <subscription> resource, or a resource subscriber may unsubscribe by deleting the <subscription> resource.

The following clauses describe procedures for Creation, Retrieval, Update and Deletion of a <subscription> resource.

10.2.11.2 Create <subscription>

This procedure shall be used to request a new subscription to be notified for the modifications of a subscribed-to resource. Generic create procedure is described in clause 10.1.1.1.

Table 10.2.11.2-1: <subscription> CREATE

<subscription> CREATE	
Associated Reference Point	MCA, Mcc and Mcc'
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: • ***cn***: The resource content shall provide the information as defined in clause 9.6.8.
Pre-Processing at Originator	According to clause 10.1.1.1 with the following additions: The Request shall address a subscribable resource. The Request shall include notificationURI(s). If the request includes notificationURI(s) which is not the Originator, the Originator should send the request as non-blocking request (see clause 8.2.2 and clause 9.6.12).
Processing at Receiver	According to clause 10.1.1.1 with the following Which is also the hosting CSE shall validate the followings: • Check if the subscribed-to resource, addressed in the ***to*** parameter in the Request, is a subscribable resource; • Check if the Originator has privileges for retrieving the subscribed-to resource; • If the notificationURI is not the Originator, the hosting CSE should send a Notify request to the notificationURI to verify this <subscription> creation request. If it cannot initiate the verification, the hosting CSE shall return unsuccessful result. If the hosting CSE initiate the verification, then it shall check if the verification result in the Notify response is successful or not; If any of the checks above fails, the hosting CSE shall send unsuccessful result to the Originator with corresponding error information. Otherwise, the hosting CSE shall create the <subscription> resource and send successful result to the Originator. If the *latestNotify* attribute is set, the hosting CSE shall assign attribute ***ec*** of value 'latest' of the notifications generated pertaining to the subscription created.
Information on Response message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: • ***cn***: URI of the created <subscription> resource, according to clause 10.1.1.1
Post-Processing at Originator	According to clause 10.1.1.1
Exceptions	According to clause 10.1.1.1

10.2.11.3 Retrieve <subscription>

This procedure shall be used to retrieve information of a subscription such as *notificationURI*, *filterCriteria*, *expirationTime*, etc. The generic retrieve procedure is described in clause 10.1.2.

Table 10.2.11.3-1: <subscription> RETRIEVE

<subscription> RETRIEVE	
Associated Reference Point	Mca, Mcc, Mcc'
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: void
Pre-Processing at Originator	According to clause 10.1.2
Processing at Receiver	According to clause 10.1.2
Information on Response message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: attributes of the <subscription> resource as defined in clause 9.6.8
Post-Processing at Originator	According to clause 10.1.2.
Exceptions	According to clause 10.1.2

10.2.11.4 Update <subscription>

This procedure shall be used to update an existing subscription, e.g. extension of its lifetime or the modification of the notificationURI. Generic update procedure is described in clause 10.1.3.

Table 10.2.11.4-1: <subscription> UPDATE

<subscription> UPDATE	
Associated Reference Point	Mca, Mcc, Mcc'
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: attributes of the <subscription> resource as defined in 9.6.8 which need be updated,
Pre-Processing at Originator	According to clause 10.1.3
Processing at Receiver	According to clause 10.1.3 If the *latestNotify* attribute is set, the hosting CSE shall assign attribute ***ec*** of value 'latest' of the notifications generated pertaining to the subscription created.
Information on Response message	According to clause 10.1.3
Post-Processing at Originator	According to clause 10.1.3
Exceptions	According to clause 10.1.3

10.2.11.5 Delete <subscription>

This procedure shall be used to unsubscribe an existing subscription. Generic delete procedure is described in clause 10.1.4.

Table 10.2.11.5-1: <subscription> DELETE

<subscription> DELETE	
Associated Reference Point	Mca, Mcc, Mcc'
Information on Request message	All parameters defined in table 8.1.2.1-1 apply
Pre-Processing at Originator	According to clause 10.1.4
Processing at Receiver	According to clause 10.1.4
Information on Response message	According to clause 10.1.4
Post-Processing at Originator	According to clause 10.1.4
Exceptions	According to clause 10.1.4

10.2.12 Notification Procedures for Resource Subscription

This procedure shall be used to notify Receiver(s) of modifications of a resource for an associated <subscription> resource and notify the <subscription> resource deletion. Also, this procedure shall be used to request resource subscription verification to Receiver(s).

Originator: The Originator can be a CSE which is hosting a <subscription> resource. The Originator shall send a Notify request containing the subscribed information following the detection of a modification of the subscribed-to resource that matches the specified *notificationCriteria* as specified in the <subscription> resource. In case of the <subscription> resource deletion, if *subscriberURI* is stored in the <subscription>, the Originator shall send Notify request to the *subscriberURI* to inform of the deletion. The notification shall include the creator of the <subscription> resource, if it is provided.

When a CSE receives a <subscription> creation request which needs verification (see clause 10.2.11.2), it can also be the Originator. The notification shall include the Originator ID of the <subscription> resource creation.

When the Originator receives unsuccessful Notify response with subscription verification failure information, the Originator shall send unsuccessful result to the corresponding <subscription> creation Originator if it does not have the <subscription> resource, otherwise the Originator may delete the corresponding <subscription> resource.

In case the notification is forwarded or aggregated by transit CSEs, the originator shall contain attributes related to notification policies such as *latestNotify* or *batchNotify* in the notification so that transit CSE is able to enforce the policy defined by the subscriber.

[10.2.12.a] Editor's Note: The notification policy enforcement between subscription resource hosting CSE and the notification target CSE need to be FFS.

Receiver: The Receiver can be AE(s) or a CSE(s) which shall be presented in the *notificationURI* attribute of the <subscription> resource. The Receiver can also be a transit CSE in case the notification is forwarded or aggregated.

If the Notify request requests resource subscription verification to the Receiver by including the Originator ID of the subscription creation, the Receiver shall check if the Notify Originator and the corresponding <subscription> creation Originator have NOTIFY privilege.

- If any of the two check is not successful, the Receiver shall return unsuccessful response to the Originator with subscription verification failure information.
- Otherwise, the Receiver shall send successful response to the Originator.

A considered Notification Request shall be sent when notification policies are satisfied.

The *expirationCounter* shall be decreased by one when the Originator successfully sends the notification request to Receiver(s). If the counter meets zero, the corresponding subscription resource is deleted.

The *batchNotify* policy is based on a minimum number of notification events and a maximum time to wait for that number of events. Notification events shall be temporarily stored for some duration (e.g. 10 minutes) or until a specified number are stored (e.g. 20 notifications) before sending - then sent when the first of these two conditions are satisfied.

The sending order is first-in first out (FIFO). *notificationEventCat* is checked at the time of batch transmission and applied to each notification in the batch. Stored notification events may be dropped according to the *notificationStoragePriority* and the *notificationCongestionPolicy* (see clause 9.6.3).

The *rateLimit* policy is based upon a maximum specified number of events (e.g. 10, 000) that can be sent within some specified *rateLimitWindow* duration (e.g. 60 seconds). The *ratelimitWindows* are sequential (not rolling). A considered Notification Request may only be sent whenever the current total number of events sent is less than the maximum number of events within the current *rateLimitWindow* duration. Notification events that do not meet this policy are temporarily stored or dropped according to the *notificationStoragePriority* and the *notificationCongestionPolicy* (see clause 9.6.3). In the event that transmission of a batch triggers rate-limiting, the transmission of the batch shall complete, and the rate limiting mechanism may hold off sending subsequent notifications within the *rateLimit* unit time in which transmissions have exceeded the allowed notifications.

The *pendingNotification* indicates the notification procedure to be taken following an unreachable period (due to lack of Originator's notification schedule or reachability schedule). When a notification is generated and the corresponding <subscription> resource has *pendingNotification*, the Originator shall verify the reachability to the Receiver with both the Originator's and Receiver's <schedule> resource. If there is reachability, the notification is immediately sent. Otherwise, the *pendingNotification* is applied. If the *pendingNotification* set to the "sendNone", the notifications shall be discarded. If it is set to the "sendLatest", the new notification shall be stored while the old one is discarded. If it is set to the "sendAllPending", the new notification shall be stored. After the reachability recovery, the processed notification by the *pendingNotification* is sent to the Receiver(s). The scope of the *pendingNotification* is the hosting CSE for the one subscription it is set in, it does not extend to transit CSEs.

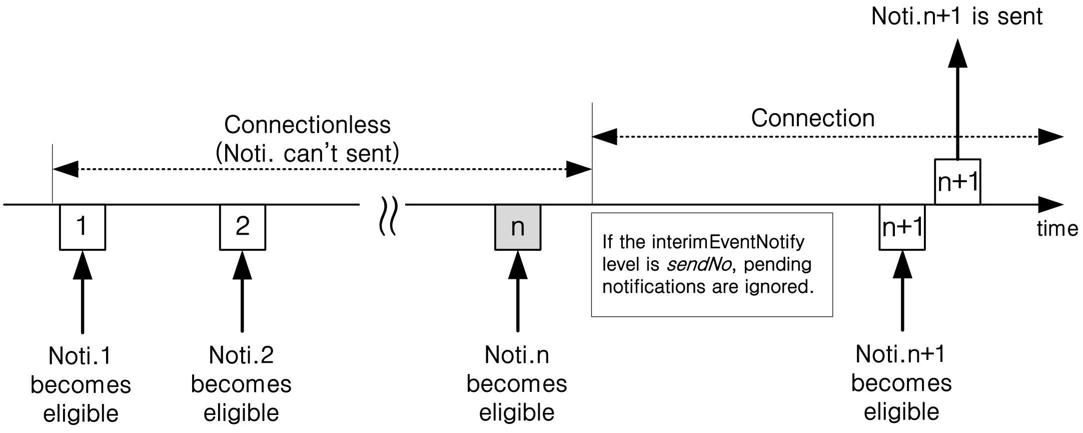


Figure 10.2.12-1: Notification Mechanism when *pendingNotification* (sendNone) is used

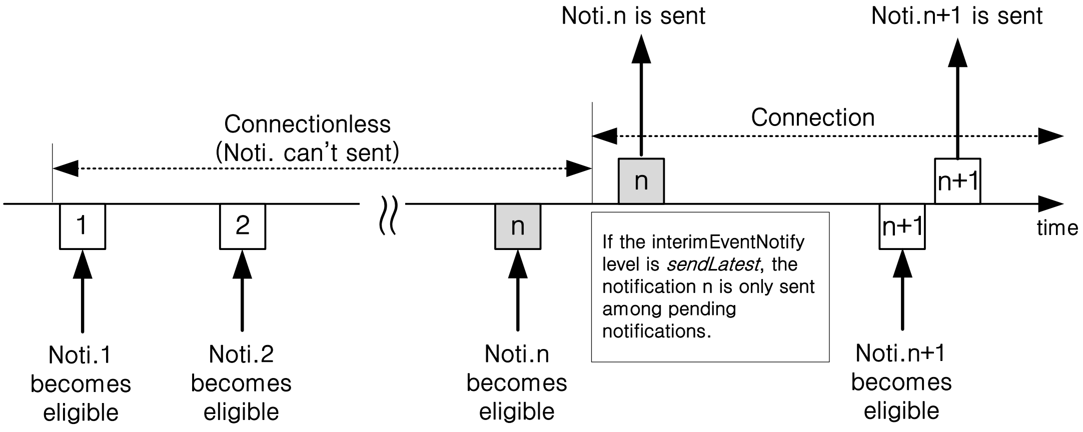


Figure 10.2.12-2: Notification Mechanism when *pendingNotification* (sendLatest) is used

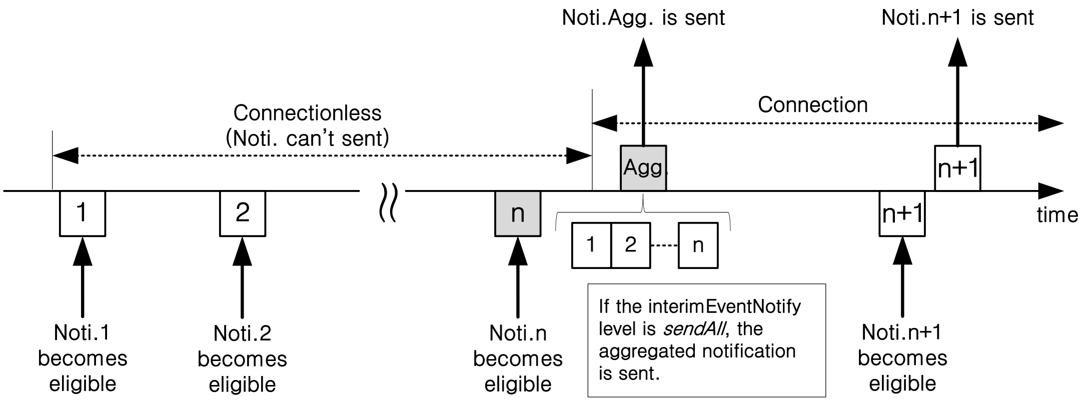


Figure 10.2.12-3: Notification Mechanism when *pendingNotification* (sendAllPending) is used

The *priorSubscriptionNotify* policy indicates the notification action to be taken following a new subscription. It enables sending of event notifications, which happened before the new subscription was created. When a new subscription is created, the most recent number "n" subscription events prior to the new subscription shall be sent to Receiver(s), if available.

The *latestNotify* indicates if the subscriber is only interested in the latest state of the subscribed-to resource. In the case the Receiver is a transit CSE which forwards or aggregates the notifications before sending to the subscriber or the other transit CSEs, upon receiving the notification with the ***ec*** set to 'latest', the Receiver shall identify the latest notification with the same subscription reference while storing the notifications locally. When the receiver as a transit CSE needs to send the pending notifications, it shall send the latest notification.

[10.2.12.b] Editor's Note: How the latestNotify is carried in the notification is FFS.

The *notificationDeliveryPriority* policy indicates how to prioritise sending of notifications when notifications need to be sent. A notification with the higher priority indicated by the *notificationDeliveryPriority* attribute shall be sent before notifications with the lower priority. In order not to constantly defer notifications with the lower priority; expiration time of the notification needs to be taken into account. In case the connectivity is lost and the notification priority exceeds a specific threshold, then if the loss of connectivity is due to the Originator's schedule then the schedule will be over-ridden and that notification shall be sent immediately.

[10.2.12.c] Editor's Note: The specific threshold mentioned above needs to be added as an attribute in the <CSEBase> resource type.

The *notificationEventCat* policy indicates an Event Category of the subscription that will be included in the notification request to be able for the Receiver to correctly handle the notification. When the *notificationEventCat* policy is not configured by the subscriber, it shall be determined as a default value by the CMDH policy.

Table 10.2.12-1: Notification Procedure

Description	
Call Flow Type	NOTIFY
Pre-Conditions	Notification is triggered regarding subscription information in a <subscription> resource
Information on Request message	***fr***: ID of the Originator ***to***: notificaitonURI specified in <subscription> resource ***cn***: • notification data that represents the modified content of subscribed-to resource may be included • subscription reference (i.e. URI of the corresponding <subscription> resource) that generates this notification shall be included • changed resource status shall be included when resourceStatus notification criteria condition is configured • monitored operation shall be included when operationMonitor notification criteria condition is configured
Information on Response message	No change from the basic procedure
Post-Conditions	None
Exceptions	None

10.2.13 Polling Channel Management Procedures

10.2.13.1 Introduction

An AE or a CSE that is request unreachable cannot receive a request from other entities directly. Instead this AE/CSE can retrieve requests that others sent to this AE/CSE once it created <pollingChannel> resource on a request reachable CSE.

This clause consist of manipulation procedures of <pollingChannel> resource, re-targeting request to <pollingChannel> and the Long Polling procedure to retrieve requests from <pollingChannel>.

10.2.13.2 Create <pollingChannel>

This procedure is used to create a <pollingChannel> resource and an AE/CSE can be an Originator. After the creation of the <pollingChannel> resource, the AE/CSE can perform Long Polling on it.

Originator: Can be an AE/CSE. If an AE is the Originator, it addresses the <AE> resource that it already created. Otherwise, if a CSE is the Originator, it addresses the <remoteCSE> resource that it already created. With ***to*** parameter addressing the <AE> or <remoteCSE> resource, the Originator sends the request to the Receiver.

Receiver: shall perform the following.

- Check if the Originator has a CREATE privilege to create <pollingChannel> resource under <AE> resource or <remoteCSE> resource. If the Originator is the AE who created the <AE> resource that the request is targeting, the request shall be granted. If the Originator is the CSE who created the <remoteCSE> resource that the request is targeting, the Request shall be granted. Otherwise, the Request will be rejected.

- Send the successful/unsuccessful Response to the Originator depending on the check above.

Table 10.2.13.2-1: <pollingChannel> CREATE

Description	
Call Flow Type	CREATE
Pre-Conditions	The Originator is request-unreachable.
Information on Request message	***op***: C ***fr***: ID of the Originator ***to***: URI of <AE> or <remoteCSE> resource ***cn***: Mandatory and/or optional attributes defined in clause 9.6.9.
Local processing on Hosting CSE	Check access control policies regarding creation of the <pollingChannel> resource.
Information on Response message	No change from the basic procedure in clause 10.1.1
Post-Conditions	None
Exceptions	No change from the basic procedure in clause 10.1.1

10.2.13.3 Retrieve <pollingChannel>

This procedure is used to retrieve a <pollingChannel> resource and an AE/CSE can be an Originator.

Originator: Sends a Request to the Receiver addressing a <pollingChannel> resource.

Receiver: Shall perform the following.

- Check if the Originator has a RETRIEVE privilege to create <pollingChannel> resource.

- Send the successful/unsuccessful response to the Originator depending on the check above.

Table 10.2.13.3-1: <pollingChannel> RETRIEVE

Description	
Call Flow Type	RETRIEVE
Pre-Conditions	None
Information on Request message	***op***: R ***fr***: ID of the Originator ***to***: URI of the <pollingChannel> resource ***cn***: Mandatory and/or optional attributes defined in clause 9.6.9.
Local processing on Hosting CSE	Check access control policies regarding retrieval of the <pollingChannel> resource
Information on Response message	NOTE: Virtual attribute *longPollingURI* is not included. No change from the basic procedure in clause 10.1.2
Post-Conditions	None
Exceptions	No change from the basic procedure in clause 10.1.2.

10.2.13.4 Update <pollingChannel>

This procedure is used to update a <pollingChannel> resource and an AE/CSE can be an Originator.

Originator: Sends a Request to the Receiver addressing a <pollingChannel> resource.

Receiver: Shall perform the following.

- Check if the Originator has a UPDATE privilege to create <pollingChannel>.
- Send the successful/unsuccessful response to the Originator depending on the check above.

Table 10.2.13.4-1: <pollingChannel> UPDATE

Description	
Call Flow Type	UPDATE
Pre-Conditions	None
Information on Request message	***op***: U ***fr***: ID of the Originator ***to***: URI of the <pollingChannel> resource ***cn***: Mandatory and/or optional attributes defined in clause 9.6.21.
Local processing on Hosting CSE	Check access control policies regarding creation of the subscription resource and sending notification to the *notificationURI.*
Information on Response message	No change from the basic procedure in clause 10.1.3
Post-Conditions	None
Exceptions	No change from the basic procedure in clause 10.1.3

10.2.13.5 Delete <pollingChannel>

This procedure is used to delete a <pollingChannel> resource and an AE/CSE can be an Originator.

Originator: Sends a Request to the Receiver addressing a <pollingChannel> resource.

Receiver: Shall perform the following.

- Check if the Originator has a UPDATE privilege to create <pollingChannel>.
- Send the successful/unsuccessful Response to the Originator depending on the check above.

Table 10.2.13.5-1: <pollingChannel> DELETE

Description	
Call Flow Type	DELETE
Pre-Conditions	None
Information on Request message	***op***: D ***fr***: ID of the Originator ***to***: URI of the <pollingChannel> resource
Local processing on Hosting CSE	Check access control policies regarding deletion of the <pollingChannel> resource
Information on Response message	No change from the basic procedure in clause 10.1.4
Post-Conditions	None
Exceptions	No change from the basic procedure in clause 10.1.4

10.2.13.6 Re-targeting Requests to Polling Channel

This procedure is used to re-target a request targeting a request-unreachable AE/CSE. When a <pollingChannel> hosting CSE receives a request to the request-unreachable AE/CSE, it internally re-targets the request to the <pollingChannel> of the AE/CSE.

It is assumed that the request-unreachable AE/CSE already set its requestReachability attribute as FALSE and created the <pollingChannel> resource. If there is no <pollingChannel> for the entity and requestReachability is FALSE, re-targeting is not performed. Re-targeted requests shall be transmitted to its target entity when it performs Long Polling on its <pollingChannel> resource.

10.2.13.7 Long Polling on Polling Channel

This procedure is originated by a request-unreachable entity to poll requests from a polling channel. Once the Originator starts Long Polling on a <pollingChannel> by sending a RETRIEVE request, the Receiver who is the <pollingChannel> hosting CSE holds the request until it has any responses to return to the Originator. If the request expires and there's no available request return, the Receiver shall send the response to inform the Originator that a new polling request should be generated again.

Originator: Sends a RETRIEVE Request to the Receiver addressing *longPollingURI* attribute of a <pollingChannel> resource.

Receiver: Shall perform the following.

- Check if the Originator is the creator of the <pollingChannel> resource.
- If creator is not present in the resource, check if the Originator ID is the same as the CSE-ID or AE-ID of the parent resource.
- If both checks are failed, the Receiver shall reject the request.
- Check if there is any request to be returned to the Originator. If there is any, the Receiver shall generate the response containing the request(s) for the Originator. If none, the Receiver shall wait for any request for the Originator to be reached at the polling channel until the request expiration time.
- Send the successful/unsuccessful response to the Originator depending on the checks above.

Table 10.2.13.7-1: Long Polling RETRIEVE

Description	
Call Flow Type	RETRIEVE
Pre-Conditions	The Originator created a <pollingChannel>
Information on Request message	***op***: R ***fr***: ID of the Originator ***to***: *longPollingURI* of the <pollingChannel> resource
Local processing on Hosting CSE	Check ownership of the <pollingChannel> resource as described above. If it is confirmed and there's available request(s) to return, the Receiver generate the response message containing the request(s).
Information on Response message	***cn***: request message(s) targeting the Originator
Post-Conditions	If the Originator receives the response from the Receiver that the Long Polling request is expired, the Originator can send a new Long Polling request.
Exceptions	If the Long Polling request is expired at the Receiver, the Receiver shall send an unsuccessful response to the Originator with the request expiry information.

10.2.14 <node> Resource Procedure

10.2.14.1 Create <node>

This procedure shall be used for creating a <node> resource.

NOTE: The creation of the <node> resource is on discretion of the Originator. In general the resource is created when the Originator is not always reachable and therefore it is convenient that the entity that the Originator is registered to is aware of the characteristic of the node, in particular the reachability schedule. It is assumed that the Originator of this <node> resource can be M2M Service Provider or the M2M Application Provider using Mca reference point. In this latter case an IN-AE adds a list of nodes which are then declared to be part of an M2M Service Subscription.

Table 10.2.14.1-1: <node> CREATE

<node> CREATE	
Associated Reference Point	Mca, Mcc and Mcc'
Information on Request message	All parameters defined in Table 8.1.2.1-1 apply with the specific details for: ***cn***: The representation of the <node> resource described in clause 9.6.18. The following attributes from clause 9.6.18 are mandatory for the request: • "*resourceType*" which shall be set to the appropriate tag that identify the <node> resource as defined in 9.6.1 NOTE: if the Originator is a CSE, it could take the information that is stored in the <node> resource under its own <CSEBase> resource and provide the information in the ***cn.***
Pre-Processing at Originator	According to clause 10.1.1.1 with the following specific processing: The Originator shall request to create a new <node> resource. The request shall address different resources depending on the type of node that is issuing the request, as described in 9.6.18. Following that, the Originator can create the child resources (defined in the 9.6.18) to acquire node specific information as occasion demands of the Originator. The Originator shall be an AE or a CSE.
Processing at Receiver	According to clause 10.1.1.1 with the following specific processing: • Check whether the provided attributes of the <node> resource represent a valid Request.
Information on Response message	All parameters defined in Table 8.1.2.1-1 apply with the specific details for: ***cn***: URI of the created <node> resource, according to clause 10.1.1.1
Post-Processing at Originator	According to clause 10.1.1.1 with the following specific processing: If the Originator is either ASN or MN-CSE registered with a CSE: The registered CSE checks the announceTo attributes whether the created <node> resource under the <CSEBase> is announced or not. If so, the created <node> resource is automatically announced by the registered CSE.
Exceptions	According to clause 10.1.1.1

10.2.14.2 Retrieve <node>

This flow is used for retrieving the attributes of a <node> resource.

Table 10.2.14.2-1: <node> RETRIEVE

<node> RETRIEVE	
Associated Reference Point	Mca, Mcc, Mcc'
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: void
Pre-Processing at Originator	According to clause 10.1.2
Processing at Receiver	According to clause 10.1.2
Information on Response message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: attributes of the <node> resource as defined in clause 9.6.6
Post-Processing at Originator	According to clause 10.1.2.
Exceptions	According to clause 10.1.2

10.2.14.3 Update <node>

This flow is used for updating the attributes and the actual data of a <node> resource and its child resources.

Table 10.2.14.3-1: <node> UPDATE

<node> DELETE	
Associated Reference Point	Mca, Mcc, Mcc’
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for ***cn***: attributes of the <node> resource as defined in 9.6.18 which need be updated, with the exception of the Read Only (RO) attributes cannot be modified
Pre-Processing at Originator	According to clause 10.1.4
Processing at Receiver	According to clause 10.1.4 with the following The Receiver shall check whether the provided attributes of the <node> resource represent a valid request for updating <node> resource
Information on Response message	According to clause 10.1.4
Post-Processing at Originator	According to clause 10.1.4
Exceptions	According to clause 10.1.4

10.2.14.4 Delete <node>

This procedure shall be used for deleting an existing <node> resource.

Table 10.2.14.4-1: <node> DELETE

<node> DELETE	
Associated Reference Point	Mca, Mcc, Mcc’
Information on Request message	All parameters defined in table 8.1.2.1-1 apply
Pre-Processing at Originator	According to clause 10.1.4
Processing at Receiver	According to clause 10.1.4
Information on Response message	According to clause 10.1.4
Post-Processing at Originator	According to clause 10.1.4
Exceptions	According to clause 10.1.4

10.2.15 Service Charging and Accounting Procedures

10.2.15.1 Introduction

This clause is informative and provides a use case example to explain how the Infrastructure Node provides statistics for AEs using the <statsConfig> and <statsCollect> resources as defined in clauses 9.6.22, 9.6.23.and 9.6.24.

10.2.15.1.1 Service Event-based Statistics Collection for Applications

Figure 10.2.15.1-1 shows an example of service layer event-based charging based on the Infrastructure Node.

Step 1-2: A statistics collection resource called <statsConfigSCA1> was created at the IN-CSE by a billing application. Note that the <statsConfig> can also be provisioned. In this use case, the <statsConfigSCA1> has the <eventConfig> sub-resource. For this specific use case, the <eventConfig> can be set as following: The "*eventID*" is set with a unique ID to differentiate from other chargeable events. The "eventType" defines what event will trigger the generation of service statistics collection record and is set to "Data Operation" for this case. "eventStart" and "eventEnd" apply to timer based event so they will not be included in this event. "transactionType" will be "RETRIEVE". "dataSize" does not apply so it is not included.

Step 3-5: In this example, AE1 already registered to IN-CSE. IN-CSE can make the statistics collection configuration accessible by AE. Based on the <statsConfigSCA1>, AE1 creates a statistics collection trigger for itself, stored in <statsCollectAE1>. AE1 will fill in the information for the collection rule. For example, it fills the "collectingEntityID" with the App-ID of AE1, and the "collectedEntityID" empty, which means to collect for any entities. "status" is set to "Active". The "statModel" is "event-based". The "subscriberID" is set to the M2M-Sub-ID for AE1. The "*eventID*" is set with the same ID value as the "*eventID*" in the <statsConfigSCA1>. This event collection trigger can be saved at the IN-CSE and IN-CSE will assign a unique ID in attribute "statsCollectID".

Step 6-8: When the configured event happens, i.e. when AE2 performed a RETRIEVE operation to the data owned by AE1 at IN-CSE, the event is recorded by IN-CSE. IN-CSE generates a service statistics collection record that sends it to AE1. AE1 can choose to use such information for its own billing.

Step 9: The AE of billing application can update or retrieve the charging policies and collection scenarios that it has the access control privilege.

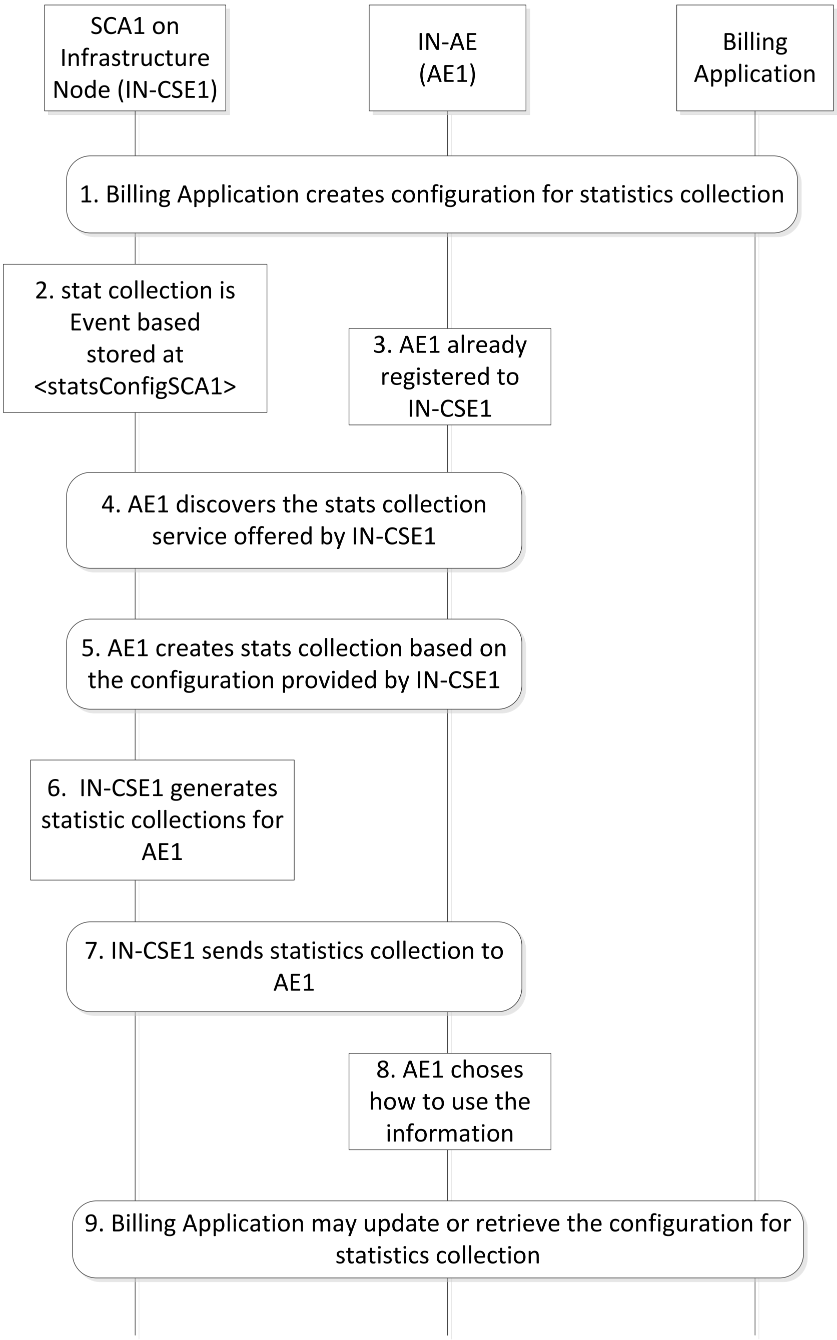

SCA1 on Infrastructure Node (IN-CSE1)
IN-AE (AE1)
Billing Application
1. Billing Application creates configuration for statistics collection
2. stat collection is Event based stored at <statsConfigSCA1>
3. AE1 already registered to IN-CSE1
4. AE1 discovers the stats collection service offered by IN-CSE1
5. AE1 creates stats collection based on the configuration provided by IN-CSE1
6. IN-CSE1 generates statistic collections for AE1
7. IN-CSE1 sends statistics collection to AE1
8. AE1 choses how to use the information
9. Billing Application may update or retrieve the configuration for statistics collection

Figure 10.2.15.1-1: Event-based Statistics Collection for Applications

10.2.15.2 CREATE <statsConfig>

This procedure shall be used for the Originator to establish a set of configurations for statistics collection at the Receiver.

The configurations shall be stored at the <statsConfig> resource and each instance of the <statsConfig> resource shall represent a specific configuration.

Originator: The Originator shall be an AE or the IN-CSE that wants to set up the statistics collection configurations. According to the definition of the <statsConfig>, the Originator shall:

- provide the sub-resource <eventConfig> for Event based statistics collection.

This sub-resource is used to define different events to be collected, each identified by a unique "*eventID*".

eventID: The "*eventID*" shall uniquely identify an event. The Originator may provide a value in the Request. The Receiver shall verify whether the ID is unique or not, and if not, provides a new value.

eventType: The "*eventType*" is a mandatory attribute. The Originator shall provide an eventType. An example of an eventType is "Data Operation", which means that chargeable events information recording is triggered when the charged entity conducts a data operation. "Data Operation" can be further refined by the "transactionType" attribute.

eventStart: The "*eventStart*" is an optional attribute. The Originator shall provide a start time for the event only if the "eventType" is set to "Timer based".

eventEnd: The "*eventEnd*" is an optional attribute. The Originator shall provide an end time for the event only if the "eventType" is set to "Timer based".

transactionType: The "*transactionType*" is an optional attribute. Examples of "*transactionType*" can be "CREATE", "RETRIEVE".

dataSize: The "*dataSize*" is an optional attribute. The Originator shall provide a value for it only if the "*eventType*" is set to "Storage based".

Receiver: The Receiver shall be an IN-CSE. The Receiver shall validate whether the Originator has proper permissions for creating a <statsConfig> resource, and whether the values provides in the Request message are valid. Upon successful validation, the Receive shall create a new <statsConfig> resource with the provided values.

Table 10.2.15.2-1: CREATE Call Flow

Description	
Call flow type	CREATE
Pre-conditions	Originator needs to create a <statsConfig> resource at the Receiver
Information on Request message	***op*: Create** ***fr***: ID of the Originator ***to*:** URI of the parent of the target resource to be created ***cn***: contain the resource representation of <statsConfig> The Originator can provide the <eventConfig> sub-resource to enable event-based configuration for statistics collection The Originator may specify other Optional parameters as discussed in clause 8.1.2 Request
Local processing on Receiver CSE	No change from the generic procedure
Generic Information on Response message	No change from the generic procedure
Post-conditions	None
Exceptions	No change from the generic procedure

10.2.15.3 RETRIEVE <statsConfig>

The RETRIEVE call flow shall be used for the Originator to retrieve the existing <statsConfig> resource from the Receiver.

Originator: The Originator shall be an AE that is allowed to retrieve configuration information available for AEs within an IN-CSE.

Receiver: The Receiver shall be the IN- CSE containing the <statsConfig> resource. The Receiver shall verify if the Originator has the READ permission on the <statsConfig> resource targeted. Upon successful validation, the Receiver CSE shall respond to the Originator with the resource representation.

Table 10.2.15.3-1: RETRIEVE Call Flow

Description	
Call flow type	RETRIEVE
Pre-conditions	Originator needs to retrieve the <statsConfig> resource at the Receiver
Information on Request message	***op*: Retrieve** ***fr*: ID of the Originator** ***to*: URI of the <statsConfig> resource or its attribute or sub-resource to be retrieved** ***cn*: void** The Originator may specify other Optional parameters as discussed in clause 8.1.2 Request
Local processing on Receiver CSE	No change from the generic procedure
Generic Information on Response message	No change from the generic procedure
Post-conditions	None
Exceptions	No change from the generic procedure

10.2.15.4 UPDATE <statsConfig>

An UPDATE procedure on the <statsConfig> resource is used for the Originator to update charging related policies at the Receiver.

Originator: The Originator shall be the AE that created the <statsConfig> resource. The same AE shall be able to update the resource.

Receiver: The Receiver shall be a CSE containing the <statsConfig> resource. The Receiver shall check if the Originator has the WRITE permission to update the addressed resource.

Table 10.2.15.4-1: UPDATE Call Flow

Description	
Call flow type	UPDATE
Pre-conditions	The Originator needs to update the charging policies that it established at the Receiver by using the <statsConfig> resource
Information on Request message	***op*:** Update ***fr*:** ID of the Originator ***to*:** URI of the <statsConfig> resource or its attribute or sub-resource to be updated ***cn*:** the Originator provides the sub-resource or attributes to be updated The Originator can update attributes under <eventConfig> to update event-based configuration for statistics collection The Originator may specify other Optional parameters as discussed in clause 8.1.2 Request
Local processing on Receiver CSE	No change from the generic procedure
Generic Information on Response message	No change from the generic procedure
Post-conditions	None
Exceptions	No change from the generic procedure

10.2.15.5 DELETE <statsConfig>

Originator: The Originator shall be the AE that created the <statsConfig> resource.

Receiver: The Receiver shall be a CSE containing the <statsConfig> resource.

Table 10.2.15.5-1: DELETE Call Flow

Description	
Call flow type	DELETE
Pre-conditions	Originator needs to DELETE the <statsConfig> resource at the Receiver
Information on Request message	***op*: Delete** ***fr*: ID of the Originator** ***to*: URI of the <statsConfig> resource to be deleted** ***cn*: void** The Originator may specify other Optional parameters as discussed in clause 8.1.2 Request
Local processing on Receiver CSE	No change from the generic procedure
Generic Information on Response message	No change from the generic procedure
Post-conditions	None
Exceptions	No change from the generic procedure

10.2.15.6 CREATE <statsCollect>

This procedure shall be used for the Originator to establish collection scenarios at the Receiver.

The collection scenarios are stored at the <statsCollect> resource. Multiple collection scenarios can be created based on one <statsConfig>.

Originator: The Originator shall be an AE that wants to set up the collection scenarios to an IN-CSE. According to the definition of the <statsCollect> resource, the Originator shall populate the attributes defined for the <statsCollect> resource except for "statsCollectID". (see clause 9.6.24). The "statsCollectID" shall be unique in the same service provider domain. To ensure the uniqueness, the IN-CSE shall create the statsCollectID.

Here are other attributes in the <statsCollect> resources:

collectingEntityID: The "*collectingEntityID*" is a mandatory attribute. The Originator shall provide the ID as defined in clause 7.1 to identify the entity that charges.

collectedEntityID: The "*collectedEntityID*" is a mandatory attribute. The Originator shall provide the ID as defined in clause 7.1 to identify the entity that being charged.

status: The "*status*" is a mandatory attribute. The Originator shall set the value for this attribute to specify the status of the rule. Examples of values can be "ACTIVE", "INACTIVE".

statModel: The "*statModel*" is a mandatory attribute. The Originator shall set the value to indicate a collection model. Examples of values can be "subscriber-based", "event-based".

subscriberID: The "*subscriberID*" is a mandatory attribute. It is the service subscriber ID for the collected entity. The Originator shall provide the "M2M-Sub-ID" as defined in clause 7.1.

collectPeriod: The "*collectPeriod*" is an optional attribute. It applies to the subscriber-based charging model. The Originator shall provide a value of the information recording collection period. An example of the value can be "monthly".

eventID: The "*eventID*" is an optional attribute. The Originator shall provide a value only if the "statModel" is "event-based". The Originator shall provide the *eventID* that is defined in the resource <statsConfig>. Since there can be many different chargeable events defined by different charging policies, the *eventID* uniquely identifies the specific chargeable event for this charging rule.

Receiver: The Receiver shall be an IN-CSE. The Receiver shall validate whether the Originator has proper permissions for creating a <statsCollect> resource. Upon successful validation, create a new <statsCollect> resource with the provided attributes. The IN-CSE shall also create a unique "statsCollectID".

Once a <statsCollect> is created and the "status" is "ACTIVE, the IN-CSE shall generate service statistics collection record when the conditions defined by the rules are met.

Table 10.2.15.6-1: CREATE Call Flow

Description	
Call flow type	CREATE
Pre-conditions	Originator needs to create a <statsCollect> resource at the Receiver
Information on Request message	***op*:** Create ***fr*:** ID of the Originator ***to*:** URI of the parent of the target resource to be created ***cn*:** contain the resource representation of <statsCollect> The Originator may specify other Optional parameters as discussed in clause 8.1.2 Request
Local processing on Receiver CSE	No change from the generic procedure
Generic Information on Response message	No change from the generic procedure
Post-conditions	None
Exceptions	No change from the generic procedure

10.2.15.7 RETRIEVE <statsCollect>

The RETRIEVE call flow shall be used for the Originator to retrieve the existing <statsCollect> resource from the Receiver.

Originator: The Originator shall be an AE that is allowed to retrieve the collection scenario information from the IN-CSE.

Receiver: The Receiver shall be the IN- CSE containing the <statsConfig> resource. The Receiver shall verify if the Originator has the READ permission on the <statsCollect> resource targeted. Upon successful validation, the Receiver CSE shall respond to the Originator with the resource representation.

Table 10.2.15.7-1: RETRIEVE Call Flow

Description	
Call flow type	RETRIEVE
Pre-conditions	Originator needs to retrieve the <statsCollect> resource at the Receiver
Information on Request message	***op*:** Retrieve ***fr*:** ID of the Originator ***to*:** URI of the <statsCollect> resource or its attribute to be retrieved ***cn*:** void The Originator may specify other Optional parameters as discussed in clause 8.1.2 Request
Local processing on Receiver CSE	No change from the generic procedure
Generic Information on Response message	No change from the generic procedure
Post-conditions	None
Exceptions	No change from the generic procedure

10.2.15.8 UPDATE <statsCollect>

An UPDATE procedure on the <statsCollect> resource is used for the Originator to update charging related policies at the Receiver.

Originator: The Originator shall be the AE that created the <statsCollect> resource. The same AE shall be able to update the resource.

Receiver: The Receiver shall be a CSE containing the <statsCollect> resource. The Receiver shall check if the Originator has the WRITE permission to update the addressed resource.

Table 10.2.15.8-1: UPDATE Call Flow

Description	
Call flow type	UPDATE
Pre-conditions	The Originator needs to update the collection scenarios that it established at the Receiver by updating the <statsCollect> resource
Information on Request message	***op*: Update ***fr*: ID of the Originator ***to*: URI of the <statsCollect> resource or its attribute to be updated ***cn*: the Originator provides the attributes to be updated The Originator may specify other Optional parameters as discussed in clause 8.1.2 Request
Local processing on Receiver CSE	No change from the generic procedure
Generic Information on Response message	No change from the generic procedure
Post-conditions	None
Exceptions	No change from the generic procedure

10.2.15.9 DELETE <statsCollect>

Originator: The Originator shall be the AE that created the <statsCollect> resource.

Receiver: The Receiver shall be a CSE containing the <statsCollect> resource.

Table10. 2.15.9-1: DELETE Call Flow

Description	
Call flow type	DELETE
Pre-conditions	Originator needs to DELETE the <statsCollect> resource at the Receiver
Information on Request message	***op*: Delete ***fr*: ID of the Originator ***to*: URI of the <statsCollect> resource to be deleted ***cn*: void The Originator may specify other Optional parameters as discussed in clause 8.1.2 Request
Local processing on Receiver CSE	No change from the generic procedure
Generic Information on Response message	No change from the generic procedure
Post-conditions	None
Exceptions	No change from the generic procedure

10.2.15.10 Service Statistics Collection Record

When the Service Statistics Collection is supported, the Information Elements shall be generated according to table 10.2.15.10-1.

The contents of each Service statistics collection record are decided by the specific collection scenario that triggered the information recording.

Table 10.2.15.10-1: Information Elements for Service Statistics Collection Record

Information Element	Mandatory / optional	Description
statsCollectID	M	It is the unique ID that identifies a specific statistics collection scenario, which triggers information recording for a specific event.
collectingEntityID	M	This is the unique ID of the entity that collects the statistics. It can be an AE-ID or CSE-ID
collectedEntityID	M	This is the unique ID of the entity whose service layer operation statistics are being collected. It can be an AE-ID or CSE-ID.
subscriberID	M	The M2M service subscription ID M2M-Sub-ID if the collected entity
event	O	This indicates a specific event type in each record, such as timer based, data operation, storage triggering. It is only present if the *statModel* is "event based"
eventStart	O	The start time for the recording the M2M event record
eventEnd	O	The end time for the recording the M2M event record
transactionType	O	Specifies the detailed type of a transaction, such as CREATE, RETRIEVE, etc.
dataSize	O	Storage Memory in Kbytes, where applicable, to store data associated events with container related operations
Vendor Specific Information	O	Defines Vendor specific information

10.2.16 <m2mServiceSubscription> Resource

10.2.16.1 Create <m2mServiceSubscription>

This flow is used for creating a <m2mServiceSubscription> resource.

Table 10.2.16.1-1: <m2mServiceSubscription> CREATE

< m2mServiceSubscription > CREATE	
Associated Reference Point	Mca, Mcc
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***to***: the Receiver or hosting CSE shall be an IN-CSE ***cn***: The resource content shall provide the information as defined in clause 9.6.19.
Pre-Processing at Originator	According to clause 10.1.1.1
Processing at Receiver	According to clause 10.1.1.1
Information on Response message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: URI of the created < m2mServiceSubscription > resource, according to clause 10.1.1
Post-Processing at Originator	According to clause 10.1.1.1
Exceptions	According to clause 10.1.1.1

10.2.16.2 Retrieve <m2mServiceSubscription>

This flow is used for retrieving the attributes of a <m2mServiceSubscription> resource.

Table 10.2.16.2-1: <m2mServiceSubscription> RETRIEVE

<m2mServiceSubscription> RETRIEVE	
Associated Reference Point	Mca, Mcc, Mcc’
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***to***: the Receiver or hosting CSE shall be an IN-CSE ***cn***: void
Pre-Processing at Originator	According to clause 10.1.2
Processing at Receiver	According to clause 10.1.2
Information on Response message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: attributes of the <m2mServiceSubscription> resource as defined in clause 9.6.19
Post-Processing at Originator	According to clause 10.1.2.
Exceptions	According to clause 10.1.2

10.2.16.3 Update <m2mServiceSubscription>

This flow is used for updating the attributes of a <m2mServiceSubscription> resource.

Table 10.2.16.3-1: <m2mServiceSubscription> UPDATE

<m2mServiceSubscription> UPDATE	
Associated Reference Point	Mca, Mcc
Information on Request message	All parameters defined in table 8.1.2.1-1 are applicable as indicate in the table with the specific details for: ***to***: the Receiver or hosting CSE shall be an IN-CSE ***cn***: attributes of the <m2mServiceSubscription> resource as defined in 9.6.19 which need be updated, with the exception of the following that cannot be modified: • “*lastModifiedTime*”, • ” subsSer&RoleList”.
Pre-Processing at Originator	According to clause 10.1.3
Processing at Receiver	According to clause 10.1.3
Information on Response message	According to clause 10.1.3
Post-Processing at Originator	According to clause 10.1.3
Exceptions	According to clause 10.1.3

10.2.16.4 Delete <m2mServiceSubscription>

This flow is used for deleting a <m2mServiceSubscription> resource residing under a <m2mServiceSubscription> resource.

Table 10.2.16.4-1: <m2mServiceSubscription> DELETE

<m2mServiceSubscription> DELETE	
Associated Reference Point	Mca, Mcc
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with specific details for: ***to***: the Receiver or hosting CSE shall be an IN-CSE
Pre-Processing at Originator	According to clause 10.1.4
Processing at Receiver	According to clause 10.1.4
Information on Response message	According to clause 10.1.4
Post-Processing at Originator	According to clause 10.1.4
Exceptions	According to clause 10.1.4

10.2.17 <nodeInfo> Resource

10.2.17.1 Create <nodeInfo>

This flow is used for creating a <nodeInfo> resource which is sub-resource of <m2mServiceSubscription> resource.

Table 10.2.17.1-1: <nodeInfo> CREATE

<nodeInfo> CREATE	
Associated Reference Point	Mca, Mcc
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***to***: the Receiver or hosting CSE shall be an IN-CSE ***cn****:* The resource content shall provide the information as defined in clause 9.6.20.
Pre-Processing at Originator	According to clause 10.1.1.1
Processing at Receiver	According to clause 10.1.1.1
Information on Response message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***cn***: URI of the created <nodeInfo> resource, according to clause 10.1.1
Post-Processing at Originator	According to clause 10.1.1.1
Exceptions	According to clause 10.1.1.1

10.2.17.2 Retrieve <nodeInfo>

This flow is used for retrieving the attributes of a <nodeInfo> resource which is sub-resource of <m2mServiceSubscription> resource.

Table 10.2.17.2-1: <nodeInfo> RETRIEVE

<nodeInfo> RETRIEVE	
Associated Reference Point	Mca, Mcc, Mcc'
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***to***: the Receiver or hosting CSE shall be an IN-CSE ***cn***: void
Pre-Processing at Originator	According to clause 10.1.2
Processing at Receiver	According to clause 10.1.2
Information on Response message	All parameters defined in table 8.1.2.1-1 apply
Post-Processing at Originator	According to clause 10.1.2.
Exceptions	According to clause 10.1.2

10.2.17.3 Update <nodeInfo>

This flow is used for updating the attributes of a <nodeInfo> resource which is sub-resource of <m2mServiceSubscription> resource.

Table 10.2.17.3-1: <nodeInfo> UPDATE

<nodeInfo> UPDATE	
Associated Reference Point	Mca, Mcc
Information on Request message	All parameters defined in table 8.1.2.1-1 are applicable as indicate in the table with the specific details for: ***to***: the Receiver or hosting CSE shall be an IN-CSE ***cn***: attributes of the <nodeInfo> resource as defined in 9.6.16 which need be updated, with the exception of the following that cannot be modified: "*lastModifiedTime*"
Pre-Processing at Originator	According to clause 10.1.3
Processing at Receiver	According to clause 10.1.3
Information on Response message	According to clause 10.1.3
Post-Processing at Originator	According to clause 10.1.3
Exceptions	According to clause 10.1.3

10.2.17.4 Delete <nodeInfo>

This flow is used for deleting a <nodeInfo> resource residing under a <m2mServiceSubscription> resource.

Table 10.2.17.4-1: <nodeInfo> DELETE

<nodeInfo> DELETE	
Associated Reference Point	Mca, Mcc
Information on Request message	All parameters defined in table 8.1.2.1-1 apply with the specific details for: ***to***: the Receiver or hosting CSE shall be an IN-CSE
Pre-Processing at Originator	According to clause 10.1.4
Processing at Receiver	According to clause 10.1.4
Information on Response message	According to clause 10.1.4
Post-Processing at Originator	According to clause 10.1.4
Exceptions	According to clause 10.1.4

10.2.18 Resource Announcement Procedure

10.2.18.1 Procedure for AE and CSE to initiate Creation of an Announced Resource

This clause describes the procedure for an AE or a CSE to initiate the creation of an announced resource.

Originator: The Originator of a Request for initiating resource announcement can be either an AE or a CSE. The Originator can initiate the creation of an announced resource during the creation of the original resource by providing the *announceTo* attribute in the CREATE Request. The Originator can initiate the creation of an announced resource by using the UPDATE Request to the *announceTo* attribute in the original resource as follows:

The Originator shall perform the following for the creation of an announced resource:

- The Originator can provide either the exact URI(s) for the announced resource or the list of CSE-IDs of the remote CSEs where the original resource needs to be announced by including such information within the *attributeTo* of the UPDATE or CREATE Request. In case of the CSE-IDs, the Receiver (original resource hosting CSE) shall decide the exact location for the announced resource at the remote CSE identified by the CSE-ID.

Receiver: Once the Originator has been successfully authorized, the Receiver (which shall be the original resource hosting CSE) shall grant the Request after successful validation of the Request.

- If the Request provides an exact URI(s) that are not already stored in the *announceTo* attribute or for newly created *announceTo* attribute, the Receiver shall announce the resource to the location identified by the URI(s) as per procedures in clause 10.2.18.4.

- If the Request provides a list of CSE-IDs of the remote CSE that are not already stored in the *announceTo* attribute of for newly created *announceTo* attribute, the Receiver shall decide the location at the remote CSE(s) identified by CSE-ID(s) for announcing the resource and shall perform the procedures in clause 10.2.18.4.

On successful completion of resource announcement as in clause 10.2.18.4, the Receiver shall performs the following:

- The Receiver shall provide the exact URI(s) of the announced resource to the Originator by updating the content of the *announceTo* attribute in the original resource and by providing it in the UPDATE or CREATE Response message depending on the Request.

Table 10.2.18.1-1: Resource Announcement: UPDATE or CREATE

UPDATE or CREATE Description	
Pre-Conditions	The Originator suggests the URI(s) or the CSE-ID(s) to which the resource will be announced.
Information on Request message	All parameters defined in table 8.1.2.1-1 are applicable as indicated in the table. In addition in case of CREATE the procedure for the specific resource is described in clause 10.2. The specific information relevant for announcing is: ***cn*****:** contains address where the resource needs to be announced (within *announceTo* attribute)
Local processing on Hosting CSE	Steps described for the Receiver of the Request as described above
Information on Response message	All parameters defined in table 8.1.2.1-1 are applicable as indicated in the table with the specific details for ***cn*** as described above for the Receiver of the Response
Post-Conditions	None
Exceptions	All exceptions described in the basic procedures (clause 10.1.1) are applicable.

10.2.18.2 Procedure at AE or CSE to Retrieve information from an Announced Resource

This clause describes the procedures for an AE or a CSE to retrieve information about an announced resource or the corresponding original resource.

Originator: The Originator of a Request for initiating retrieval of information about a resource can be either an AE or a CSE. The Originator initiates this procedure by using RETRIEVE Request. Clause 8.1.2 specifies the information to be included in the Request message. Specifically, the ***to*** parameter is set to the address (e.g. URI) of the announced resource to be retrieved. If a specific attribute is to be retrieved, the address of such attribute is included in the ***to*** parameter. The Originator can request retrieval of the original resource by targeting the announced resource at the hosting CSE by setting the ***rc*** parameter to the "original-resource".

The Originator shall perform as follows for retrieving information from an announced resource.

- The Originator shall set the ***to*** parameter in the RETRIEVE Request to the address (e.g. URI) of the resource or attribute to be retrieved.

- The Originator can request retrieval of information from an announced resource at the hosting CSE. Optionally the Originator can specify one of the values for the optional ***rc*** parameter.

Receiver: Once the Originator has been successfully authorized, the Receiver (hosting CSE) shall grant the Request after successful validation of the Request.

- If ***rc*** request message parameter set to "original-resource" is included in the Request message, the Receiver shall provide the representation of the original resource indicated by the *link* attribute in the announced resource; otherwise the information for the announced resource itself shall be retrieved according to the request parameters, as described in clause 8.1.2. The Receiver shall retrieve the original resource to return the representation of the original resource to the Originator.

- For a successful match, information from the identified announced resource (at hosting CSE) shall be returned to Originator via RETRIEVE Response.

Table 10.2.18.2-1: Resource Retrieval at local CSE: RETRIEVE

RETRIEVE Description	
Pre-Conditions	For RETRIEVE operation, the Originator knows the exact URI of the resource or the attribute that needs to be retrieved.
Information on Request message	All parameters defined in table 8.1.2.1-1 are applicable as indicated in the table with the specific details for: ***rc*****:** either: • not present • set to "announced-resource" • set to "links" ***cn*****:** void
Local processing on Hosting CSE	Steps as described by the basic procedure (clause 10.1.2) apply. In addition the steps described above for the Receiver of the Request apply.
Information on Response message	All parameters defined in table 8.1.2.1-1 are applicable as indicated in the table with the specific details for: ***cn*****:** as described above for the Receiver of the Response
Post-Conditions	None
Exceptions	All exceptions described in the basic procedure (clause 10.1.2) are applicable.

10.2.18.3 Procedure for AE and CSE to initiate Deletion of an Announced Resource

This clause describes the procedure for an AE or a CSE (not the original resource hosting CSE) to initiate the deletion of an announced resource.

Originator: The Originator of a Request for initiating resource de-announcement can be either an AE or a CSE The Originator can request to initiate the deletion of an announced resource by using UPDATE Request to the *announceTo* attribute at the original resource hosting CSE. Resource de-announcement (deletion) shall also be performed when the Originator deletes the original resource at the original resource hosting CSE by using DELETE Request.

The Originator shall perform as follows for the deletion of an announced resource.

- The Originator shall update the *announceTo* attribute at the original resource hosting CSE by providing new content of the *announceTo* which does not include the URI of the announced resource that needs to be de-announced (deleted) in the UPDATE operation.

- The Originator shall update the original resource in the hosting CSE by providing new representation of the resource that does not contain the *announceTo* attribute for all the previously announced resources by using an UPDATE operation

- For DELETE Request, the Originator shall include the URI of the original resource hosting CSE that needs to be deleted, in the DELETE Request.

Receiver: Once the Originator has been successfully authorized, the Receiver (which shall be the original resource hosting CSE) shall grant the Request after successful validation of the Request. The Receiver shall be the resource hosting CSE. On receiving the UPDATE or DELETE Request, the Receiver shall perform as follows:

- For UPDATE Request, the Receiver shall request to delete the announced resource whose URI is not included in the *announceTo* attribute of the request as per procedures in clause 10.2.18.5For DELETE Request, the Receiver shall request to delete all announced resources in the *announceTo* attribute as per procedures in clause 10.2.18.5.

Receiver after completion of the resource de-announcement procedure in clause 10.2.18.5:

On successful completion of resource de-announcement procedure in clause 10.2.18.5, the Receiver knows that the announced resource has been deleted.

- The Receiver shall provide confirmation of resource de-announcement to the Originator, and the content of the updated announceTo attribute shall be provided to the Originator to indicate the successfully deleted announced resource if the *announceTo* attribute is not deleted by the Originator.

Table 10.2.18.3-1: Resource De-Announcement: UPDATE and DELETE

UPDATE and DELETE Description	
Pre-Conditions	None
Information on Request message	All parameters defined in table 8.1.2.1-1 are applicable as indicated in the table with the specific details for. ***cn*****:** contains • Either the modified *announceTo* attribute • Or a complete representation of the original resource without the *announceTo* attribute In case of DELETE, the generic procedure is described in clause 10.1.4 ***cn*****:** void
Local processing on Hosting CSE	The basic procedure (clauses 10.1.3 for UPDATE and 10.1.14 for DELETE) still applies. In addition the steps described above for the Receiver of the Request also apply
Information on Response message	The basic procedure (clauses 10.1.3 for UPDATE and 10.1.4 for DELETE) still applies. In addition the steps described above for the Receiver of the Request also apply
Post-Conditions	None
Exceptions	All exceptions described in the basic procedure (clause 10.1.2) are applicable for UPDATE operation. All exceptions described in the basic procedure (clause 10.1.4) are applicable for DELETE operation

10.2.18.4 Procedure for original resource Hosting CSE to Create an Announced Resource

This clause explains the resource announcement procedure that shall be used by the original resource hosting CSE to announce the original resource to the remote CSE.

Originator: The Originator of this Request shall be the original resource hosting CSE. The Originator shall request to create the announced resource by using CREATE Request. The Request shall provide the information as follows:

- Attributes marked with **MA** and attributes marked with **OA** that are included in the *announcedAttribute* attribute at the original resource shall be provided in the CREATE Request. Such attributes shall have the same value as for the original resource.

- Attributes marked with **NA** for the original resource shall not be provided in the CREATE Request.

- The *link* attribute of the announced resource shall have the URI for the original resource

- The *labels* attribute of the announced resource shall have the same value with the original resource

- The *accessControlPolicyIDs* attribute shall always be provided in the CREATE Request even if it is not present in the original resource. In this case the original resource shall include *accessControlPolicyIDs* from its parent resource or from the local policy at the original resource, as needed.

  accessControlPolicyIDs and *labels* attributes, if present at the original resource, shall be marked **MA** and shall be provided by the original resource hosting CSE in the CREATE Request. Such attributes shall have the same value at the original resource and at the announced resource(s).

Receiver: Once the Originator has been successfully authorized, the Receiver shall grant the Request after successful validation of the Request. The Receiver shall perform as follows:

- The created announced resource shall include the common attributes specified in clause 9.6.25.1. The created announced resource shall contain the additional attributes that are provided by the Originator; i.e. attributes marked with **MA** and the attributes marked with **OA** that are included in the *announcedAttribute* attribute.

- The created announced resource shall set the *accessControlPolicyIDs* attribute to the value received in the Request message, and shall set the *labels* attribute (if present) and the *link* attribute to the value received in the Request message.

- Respond to the Originator with the CREATE Response. In this Response, the URI of the successfully announced resource shall be provided.

The Originator after receiving the Response from the Receiver shall perform the following steps:

- If the announced resource has been successfully created, the *announceTo* attribute of the original resource shall be updated to include the URI(s) for the successfully announced resource at the Receiver. The *announcedAttribute* attribute shall be updated as well to represent the successfully announced attributes as received in the Response.

- For the attributes marked as **MA** and for the attributes marked as **OA** that are included in the *announcedAttribute* attribute, the Originator shall further take the responsibility to keep their values synchronized at the announced resource by using UPDATE operation (clause 10.1.3).

Table 10.2.18.4-1: Resource Hosting CSE to Announce Resource: CREATE

CREATE Description	
Pre-Conditions	None
Information on Request message	All parameters defined in table 8.1.2.1-1 are applicable as indicated in the table with the specific details for the following attributes from clause 9.6.25 are mandatory for the request: ***cn*****:** contains **MA** attributes and **OA** attributes that are included in *announcedAttribute* attribute The following attributes from clause 9.6.25 are mandatory for the Request: • *resourceType* which shall be set to the appropriate tag that identify the <Annc> resource; • *accessControlPolicyIDs* which it contains the value provided by the Originator, as described above. • *expirationTime* provided by the originator equal to the one of the original resource • *labels* provided by the originator equal to the one of the original resource, if present All attributes from the original resource that are present in the *announcedAttribute* shall be created in the announced resource and the value of these attribute shall be equal to the one from the original resource.
Local processing on Hosting CSE	The basic procedure (clause 10.1.1) and the steps described above for the Receiver of the CREATE Request apply
Information on Response message	All parameters defined in table 8.1.2.1-1 are applicable as indicated in the table with the specific details for: ***cn***: URI where the announced resource is created according to clause 10.1.1
Post-Conditions	Steps described for the 'Originator on receiving the Response from the Receiver' as described above
Exceptions	All exception described in the basic procedure (clause 10.1.1) are applicable.

10.2.18.5 Procedure for original resource Hosting CSE to Delete an Announced Resource

This clause explains the procedure for deleting an announced resource (i.e. the resource de-announcement). This procedure shall be used by the original resource hosting CSE for deleting the announced resource that resides at the remote CSE.

Originator: The Originator of this Request shall be the original resource hosting CSE. The Originator shall request to delete an announced resource by using the DELETE Request in which the ***to*** parameter provides a URI that identifies the announced resource to be deleted.

Receiver: Once the Originator has been successfully authorized, the Receiver shall grant the Request after successful validation of the Request.

- Delete the announced resource identified by the ***to*** parameter in the Request.

- Respond to the Originator with the appropriate DELETE Response.

Originator after receiving the Response from the Receiver:

- If the announced resource is successfully deleted, the *announceTo* attribute in the original resource shall be updated to delete the URI for the deleted announced resource.

Table 10.2.18.5-1: Resource Hosting CSE to Announce Resource: DELETE

DELETE Description	
Pre-Conditions	None
Information on Request message	All parameters defined in table 8.1.2.1-1 are applicable as indicate in the table. ***op*: D** ***fr***: Identifier of the CSE that initiates the Request ***to***: the URI where announced resource needs to be deleted
Local processing on Hosting CSE	No change from the basic procedure (clause 10.1.4). Steps for the Receiver as described above
Information on Response message	No change from the basic procedure (clause 10.1.4)
Post-Conditions	Steps for the 'Originator on receiving the Response from the Receiver' as described above
Exceptions	All exception described in the basic procedure (clause 10.1.4) are applicable.

10.2.18.6 Procedure for AE and CSE to initiate the Creation of an Announced Attribute

This clause describes the procedure for an AE and CSE (not the original resource hosting CSE) to initiate the creation of an announced attribute (attribute announcement).

Originator: The Originator of a Request, for initiating attribute announcement, can be either AE or CSE (not the original resource hosting CSE). The Originator shall request attribute announcement by updating the *announcedAttribute* attribute at the original resource:

- The Originator shall update the *announcedAttribute* attribute at the original resource by adding the attribute name for the attribute that needs to be announced by using the UPDATE Request. Only the attributes marked with OA can be announced to remote announced resources.

Receiver: Once the Originator has been successfully authorized, the Receiver, which shall be the original resource hosting CSE, shall grant the Request after successful validation of the Request.

- The attributes received in the Request, which are not marked as OA, are invalid.

- The attributes received in the Request, which are not present in the original resource structure, are invalid.

- If some attributes received in the Request do not already exist in the *announcedAttribute* attribute, the Receiver shall announce such attributes to all announced resources listed in the *announceTo* attribute as per procedures in clause 10.2.18.8.

On successful announcement of attributes as per procedures in clause 10.2.18.8, the Receiver shall perform the following:

- The Receiver shall respond to the requesting AE/CSE with UPDATE Response as specified in clause 10.1.3. The content of the announced attributes can be provided in such Response.

Table 10.2.18.6-1: Creating Announced Attributes

Description	
Call Flow Type	UPDATE
Pre-Conditions	A resource has already been announced.
Information on Request message	Parameters defined in table 8.1.2.1-1 that are applicable for UPDATE. ***cn*** parameter includes the names of the attributes to be announced.
Local processing on Hosting CSE	Steps described for the Receiver of the Request as described above
Information on Response message	Parameters defined in table 8.1.2.1-1 that are applicable.
Post-Conditions	None
Exceptions	All exceptions described in the basic procedures in clause 10.1.3 are applicable.

10.2.18.7 Procedure for AE and CSE to initiate the Deletion of an Announced Attribute

This clause describes the procedure for an AE and CSE (not the original resource hosting CSE) to initiate the deletion of announced attributes (attribute de-announcement).

Originator: The Originator of a Request, for initiating attribute de-announcement, can be either AE or CSE (not the original resource hosting CSE). The Originator shall request attribute de-announcement by updating the *announcedAttribute* attribute at the original resource as follows:

- The Originator shall update the *announcedAttribute* attribute at the original resource by deleting the attribute name for the attribute that needs to be de-announced by using the UPDATE Request. Only the attributes marked with OA can be de-announced to remote announced resources.

Receiver: Once the Originator has been successfully authorized, the Receiver, which shall be the original resource hosting CSE, shall grant the Request after successful validation of the Request.

- The attributes received in the Request, which are not marked as OA, are invalid.

- If some attributes that exist in the *announcedAttribute* attribute are not received in the Request (i.e. attributes that need to be deleted by the UPDATE Request), the Receiver shall de-announce such attributes to all announced resources listed in the *announceTo* attributes as per procedure in clause 10.2.18.9.

On successful de-announcement of all attributes as per procedures in clause 10.2.18.9, the Receiver shall perform the following:

- The Receiver shall respond to the requesting AE/CSE with UPDATE Response as specified in clause 10.1.3. The names of the de-announced attributes can be provided in such Response.

Table 10.2.18.7-1: Deleting Announced Attributes

Description	
Call Flow Type	UPDATE
Pre-Conditions	A resource has already been announced.
Information on Request message	Parameters defined in table 8.1.2.1-1 that are applicable for UPDATE.. ***cn*** parameter does not include the names of the attributes to be de-announced.
Local processing on Hosting CSE	Steps described for the Receiver of the Request as described above
Information on Response message	Parameters defined in table 8.1.2.1-1 that are applicable.
Post-Conditions	None
Exceptions	All exceptions described in the basic procedures in clause 10.1.3 are applicable.

10.2.18.8 Procedure for original resource Hosting CSE for Announcing Attributes

This clause describes procedure that shall be used by the original resource hosting CSE to create announced attributes at the remote announced resources (i.e. the attribute announcement).

Originator: The Originator of this Request shall be the original resource hosting CSE. The Originator shall request to create attributes at the announced resources by using the UPDATE Request as specified in clause 10.1.3.

Receiver: Once the Originator has been successfully authorized, the Receiver (CSE hosting announced resource) shall grant the Request after successful validation of the Request. The Receiver shall perform as follows:

- Create announced attributes at the announced resource as per procedures in clause 10.1.3. The initial value for the announced attribute shall use the same value as with the original resource.

- Respond to the Originator with UPDATE Response as in clause 10.1.3.

Originator after receiving the Response from the Receiver shall perform the following steps: If the announced attributes have been successfully created, the *announcedAttribute* attribute shall be updated to include the attribute names for the successfully announced attributes.

- For the newly announced attributes in the *announcedAttribute* attribute, the Originator shall take the responsibility to keep their values synchronized at the announced resources by using UPDATE operation as in clause 10.1.3.

Table 10.2.18.8-1: Original Resource Hosting CSE to Announce Attribute: UPDATE

Description	
Call Flow Type	UPDATE
Pre-Conditions	Only the attribute marked with OA can be announced.
Information on Request message	Information described for the Originator of the UPDATE Request as in clause 10.1.3. ***cn*** parameter includes the names of the attributes to be created and their values.
Local processing on Hosting CSE	Steps described for the Receiver of the UPDATE Request as described in clause 10.1.3.
Information on Response message	Steps described for the Receiver of the UPDATE Request as described in clause 10.1.3.
Post-Conditions	Steps described for the 'Originator on receiving the Response from the Receiver' as described in clause 10.1.3
Exceptions	All exceptions described for the basic procedure in clause 10.1.3.

10.2.18.9 Procedure for original resource Hosting CSE for De-Announcing Attributes

This clause describes procedure that shall be used by the original resource hosting CSE to remove announced attributes at remote announced resources (i.e. the attribute de-announcement).

Originator: The Originator of this Request shall be the original resource hosting CSE. The Originator shall request to delete announced attributes by using the UPDATE Request as specified in clause 10.1.3. The ***cn*** parameter in the UPDATE Request provides the attribute names for the announced attributes to be deleted with their values set to NULL.

Receiver: Once the Originator has been successfully authorized, the Receiver (CSE hosting announced resource) shall grant the Request after successful validation of the Request. The Receiver shall perform as follows:

- Delete the de-announced attributes identified by the ***cn*** parameter in the UPDATE Request as per procedures in clause 10.1.3.

- Respond to the Originator with the appropriate UPDATE Response as in clause 10.1.3.

Originator after receiving the Response from the Receiver shall perform the following steps: If the attributes have been successfully removed, the *announcedAttribute* attribute shall be updated so as to remove the attribute names for the successfully de-announced attributes.

Table 10.2.18.9-1: Original Resource Hosting CSE to De-Announce Attribute: UPDATE

Description	
Call Flow Type	UPDATE
Pre-Conditions	Only the attribute marked with OA can be de-announced
Information on Request message	Information described for the Originator of the UPDATE Request as in clause 10.1.3. ***cn*** parameter includes the names of the attributes to be deleted with their values set to NULL.
Local processing on Hosting CSE	Steps described for the Receiver of the UPDATE Request as described in clause 10.1.3.
Information on Response message	Steps described for the Receiver of the UPDATE Request as described in clause 10.1.3.
Post-Conditions	Steps described for the 'Originator on receiving the Response from the Receiver' as described in clause 10.1.3.
Exceptions	All exceptions described for the basic procedure in clause 10.1.3.

10.2.19 <contentInstance> Resource

10.2.19.1 Introduction

This clause describes the management procedures for the <contentInstance> resource. Since<contentInstance> resource is immutable once created, there is no procedure for updating it.

10.2.19.2 <contentInstance> CREATE

This flow is used for creating a <contentInstance> resource.

Originator: the Originator may be an Application Entity or a CSE.

Receiver: Creation shall be allowed if the Originator is authorised to create a child resource according to the access control policies defined for the parent <container> resource. If the newly created <contentInstance> resource violates any of the policies defined in the parent <container> resource (e.g. maxNrOfInstances or maxByteSize), then the oldest <contentInstance> resources shall be removed from the <container> to enable the creation of the new *contentInstance*.

Then the Receiver shall change the "latest" attribute of the <container> resource to refer to the newly created <contentInstance> resource.

Table 10.2.19.2-1: <contentInstance> CREATE

CREATE: Description	
Pre-Conditions	The Originating AE or CSE has to be successfully authenticated and access to the Receiver has been granted.
Information on Request message	All parameters defined in table 8.1.2.1-1 are applicable as indicated in the table with the specific details for the following attributes from section 9.6.7 are mandatory for the request: ***fr:*** Identifier of the AE or the CSE that initiates the Request ***to:*** The URI of the <container> resource where the <contentInstance > resource is intended to be created, ***cn:*** The resource content shall provide the information as defined in section 9.6.7. The following attributes from section 9.6.7 are mandatory for the Request: • “contentSize” • “content”
Local processing on Hosting CSE	No change from the basic procedure (clause 10.1.1)
Information on Response message	All parameters defined in table 8.1.5-1 are applicable as indicated in the table with the specific details for: ***cn:*** URI of the created <contentInstance> resource according to section 10.1.1.
Post-Conditions	None
Exceptions	All exceptions described in the basic procedures (section 10.1.1) are applicable.

10.2.19.3 <contentInstance> RETRIEVE

This flow is used for retrieving the attributes of a <contentInstance> resource.

Originator: the Originator may be an Application Entity or a CSE.

Receiver: Once the Originator has been successfully authenticated and access to the Receiver has been granted (as described in section 11.3.4), the Receiver shall retrieve the <contentInstance> resource.

Table 10.2.19.3-1: <contentInstance> RETRIEVE

RETRIEVE: Description	
Pre-Conditions	The targeted <contentInstance> has been created.
Information on Request message	All parameters defined in table 8.1.2.1-1 are applicable as indicated in the table with the specific details for: ***cn:*** void
Local processing on Hosting CSE	No change from the basic procedure (clause 10.1.2)
Information on Response message	All parameters defined in table 8.1.5-1 are applicable as indicated in the table with the specific details for: ***cn:*** attributes of the <contentInstance> resource as defined in clause 9.6.7
Post-Conditions	None
Exceptions	All exceptions described in the basic procedure (clause 10.1.2) are applicable.

10.2.19.4 <contentInstance> DELETE

This flow is used for deleting a <contentInstance> resource residing under a <container> resource.

Originator: the Originator may be an Application Entity or a CSE.

Receiver: Once the Originator has been successfully authenticated and access to the Receiver has been granted (as described in section 11.3.4), the Receiver shall delete the <contentInstance> and update the "latest" attribute of the parent <container> resource.

Table 10.2.19.4-1: <contentInstance> DELETE

DELETE: Description	
Pre-Conditions	The targeted <contentInstance> has been created.
Information on Request message	All parameters defined in table 8.1.2.1-1 are applicable as indicated in the table.
Local processing on Hosting CSE	No change from the basic procedure (clause 10.1.4).
Information on Response message	No change from the basic procedure (clause 10.1.4)
Post-Conditions	None
Exceptions	All exceptions described in the basic procedure (section 10.1.4) are applicable.

10.2.20 <request> Resource Procedure

10.2.20.1 Create <request>

As specified in Clause 9.6.12, creation of a <request> resource can only be done on a Receiver CSE implicitly when a Registree AE or a Registree/Registrar CSE of the Receiver CSE issues a request to the Receiver CSE for targeting any other resource type or requesting a notification in non-blocking mode. Therefore, the creation procedure of a <request> resource cannot be initiated explicitly by an Originator. Creation of a <request> procedure is processed on a Receiver CSE to support a standardized interface to information representing the context and current status of a non-blocking request issued by a Registree AE or a Registree/Registrar CSE to the Receiver CSE at an earlier time

The specific condition when a <request> resource is created is as follows: When an AE or CSE issues a request for targeting any other resource type or requesting a notification in non-blocking mode , i.e. the ***rt*** parameter of the request is set to either ‘nonBlockingRequestSynch’ or ‘nonBlockingRequestAsynch’, and if the Registrar CSE of the Originator supports the <request> resource type as indicated by the ‘supportedResourceType’ attribute of the <CSEBase> resource representing the Registrar CSE of the Originator, the Registrar CSE of the Originator shall create an instance of <request> to capture and expose the context of the associated non-blocking request

A request message for creating a <request> resource is not Applicable. A <request> resource shall not be created explicitly. The Receiver CSE of a non-blocking Request that was issued by either:

- a Registrar AE of the Receiver CSE or
- a Registree/Registrar CSE of the Receiver CSE

is the Hosing CSE for the <request> resource that shall be associated with the non-blocking request.

The hosting CSE shall follow the procedure outlined in this clause

Step 001: The Receiver shall:

2) Assign an identifier to the <request> resource to be created.

3) Assign a value to the following common attributes specified in clause 9.6.1:

    a. parentID;

    b. creationTime;

    c. expirationTime: The Receiver shall assign a value that is consistent with the ***rqet***, ***rc***, ***rset*** and ***rp*** parameters effective for the associated non-blocking request that implied the creation of this <request> resource (within the restriction of the Receiver policies). If a value consistent with the ***rqet***, ***rc***, ***rset*** and ***rp*** parameters effective for the associated non-blocking request that implied the creation of this <request> resource cannot be supported, due to either policy or subscription restrictions, the Receiver will assign a new value.

    d. lastModifiedTime: which is equals to the creationTime;

    e. stateTag;

    f. accessControlPolicyIDs: Populate with one ID of an <accessControlPolicy> that contains the following:

        i. In the ‘privileges’ child resource

            1. Allow RUD operations to <request> resource being created to the hosting CSE

            2. Allow RD operations to this <request> resource being created to the Originator of the associated non-blocking request, i.e. the value of the parameter ***fr*** in the associated non-blocking request that implied the creation of this <request> resource

        ii. In the 'selfPrivileges' child resource

            1. Allow U operations the parent <accessControlPolicy> resource to the Originator of the associated non-blocking request, i.e. the value of the parameter fr in the associated non-blocking request that implied the creation of this <request> resource

4) Assign any other RO (Read Only) attributes of <request> resource type within the restriction of the Receiver policies:

    a. operation: Value of the parameter ***op*** in the associated non-blocking request that implied the creation of this <request> resource;

    b. target: Value of the parameter ***to*** in the associated non-blocking request that implied the creation of this <request> resource;

    c. originator: Value of the parameter ***fr*** in the associated non-blocking request that implied the creation of this <request> resource;

    d. requestIdentifier: Value of the parameter ***ri*** in the associated non-blocking request that implied the creation of this <request> resource;

e. metaInformation: The content of this attribute is set to information in any other optional parameters described in clause 8.1. given in the associated non-blocking request that implied the creation of this <request> resource;

f. content: Value of the parameter ***cn*** – if any – in the associated non-blocking request that implied the creation of this <request> resource;

g. requestStatus: Information on the initial status of the associated non-blocking request that implied the creation of this <request> resource. The initial value of this attribute shall be identical to the status that is contained in the Acknowledgement response message of the associated non-blocking request. Possible values for status information contained in this attribute are specified in TS-0004. The value of this attribute is subject to changes according to the progress in processing of the non-blocking request that implied the creation of this <request> resource;

h. operationResult: Initially Empty. This attribute will be used for representing the result of the originally requested operation – if any – in line with the ***rc*** parameter in the associated non-blocking request that implied the creation of this <request> resource.

Step 002: The Receiver shall create the <request> resource.

Table 10.2.20.1-1: <request> CREATE

<request> CREATE	
Associated Reference Point	None
Information on Request message	Not applicable. For <request> resources, explicit creation via a Request message shall not be supported
Pre-Processing at Originator	Not applicable. There is no Originator. <request> resources are only created implicitly.
Processing at Receiver	Different to the non-registration CREATE procedure described in clause 10.1.1.1, see outlined steps described in the present clause above.
Information on Response message	Not applicable. Since <request> resources shall not be created explicitly, no response messages will be sent after creation. However, the address of a <request> resource will be passed back as a reference to the Originator of an associated non-blocking request that implied the creation of this <request> resource.
Post-Processing at Originator	None
Exceptions	None

10.2.20.2 Retrieve <request>

This procedure shall be used for retrieving the attributes of a <request> resource.

Table 10.2.20.2-1: <request> RETRIEVE

<request> RETRIEVE	
Associated Reference Point	Mca, Mcc, Mcc’
Information on Request message	All parameters defined in Table 8.1.2.1-1 apply with the specific details for: ***cn***: void
Pre-Processing at Originator	According to clause 10.1.2 with the following specific processing: Originator needs to retrieve information about an associated previously issued non-blocking request.
Processing at Receiver	According to clause 10.1.2 with the following specific processing: The Receiver shall provide the content of the addressed <request> resource or the addressed attributes thereof.
Information on Response message	All parameters defined in Table 8.1.2.1-1 apply with the specific details for: ***cn***: attributes of the <request> resource as defined in clause 9.6.12
Post-Processing at Originator	According to clause 10.1.2
Exceptions	According to clause 10.1.2 According to clause 10.1.1.1 with the following: • The Originator CSE is not authorized to retrieve the <request> resource or the addressed parts of it • The addressed <request> resource does not exist

10.2.20.3 Update <request>

For a <request> resource explicit update requests shall not be supported. Changes in the attributes of a <request> resource can only be done by the hosting CSE due to changes of the status of the associated non-blocking request that implied the creation of this <request> resource or due to reception of an operation result in response to the associated non-blocking request that implied the creation of this <request> resource.

10.2.20.4 Delete <request>

This procedure shall be used for deleting an existing <request> resource. Deletion of an existing <request> resource shall terminate any further processing of an associated pending non-blocking request that implied the creation of this <request> resource if the pending request was not already completed or forwarded.

Table 10.2.20.4-1: <request> DELETE

<request> DELETE	
Associated Reference Point	Mca, Mcc, Mcc’
Information on Request message	All parameters defined in Table 8.1.2.1-1 apply
Pre-Processing at Originator	According to clause 10.1.4 with the following Originator needs to cancel a previously issued non-blocking request that is still pending, i.e. it has not yet been completed or Originator needs to remove the <request> resource representing the context of an already completed non-blocking request.
Processing at Receiver	According to clause 10.1.4 • Receiver CSE checks if the associated non-blocking request process is still pending. If so, it stops that request process • Receiver CSE removes the addressed <request> resource
Information on Response message	According to clause 10.1.4 with the following specific information: Successful Response message indicating that the associated non-blocking request process was stopped as requested or the context of an already completed associated non-blocking request was deleted.
Post-Processing at Originator	According to clause 10.1.4
Exceptions	According to clause 10.1.4 with the following • The Originator CSE is not authorized to delete the <request> resource • The addressed <request> resource does not exist

11 Trust Enabling Architecture

The Trust Enabling Architecture serves the purpose of establishing security and trust between all parties involved in the M2M ecosystem. It comprises the following infrastructure functions which may be external to the CSEs:

- M2M Enrolment functions, which manage the enrolment of M2M Nodes and M2M applications for access to M2M Services provided by an M2M Service Provider.
- M2M Authentication functions, in charge of identification and authentication of CSEs and AEs.
- M2M Authorization functions, which handle authorization requests to access resources.

The above functionalities are assumed to be operated by trusted parties (generally M2M Service Providers but possibly trusted third parties). These functions are detailed in TS-0003 [1].

11.1 Enrolling M2M Nodes and M2M Applications for oneM2M Services

Though M2M nodes in the field domain are assumed to communicate without human involvement, individuals or organizations remain responsible for setting the access control policies used to authorize their M2M nodes to access M2M services. In the following text, M2M Nodes refers to M2M field nodes.

In particular, individuals or organizations acquiring M2M nodes can subscribe to a contract with an M2M Service provider (M2M Service Subscription) under which they enrol their M2M nodes (e.g. using identifiers pre-provisioned on the nodes, such as Node-ID). This in turn may require an M2M Service provisioning step (including Security provisioning) that takes place on the target M2M nodes themselves, for which interoperable procedures are specified by oneM2M (see clause 11.2.1). Following M2M service provisioning, the nodes can be identified and authenticated by an M2M Authentication Function for association with an M2M Service Subscription, whose properties reflect the contractual agreement established between their owner and the M2M Service Provider.

Similarly, it shall be possible for an M2M Service Provider to mandate that application accessing M2M services be provisioned with security credentials used to authorize specific operations to instantiated applications (see clause 11.2.2). This step facilitates the deployment and management of applications that are instantiated in great numbers, as it enables all instances of an application to be managed through common security policies that are set once for all. It also enables keeping control over applications issued by untrusted sources.

The above steps may be delegated to an M2M trust enabler, when this role is not assumed by the M2M Service Provider.

11.2 M2M Initial Provisioning Procedures

11.2.1 M2M Node Enrolment and Service Provisioning

M2M service provisioning is the process by which M2M nodes are loaded with the specific information needed to seamlessly access the M2M Services offered by an M2M Service Provider. This is an initial step performed only when an M2M node is enroled for using the M2M services of an M2M Service Provider. Though this process can be performed during device manufacturing, there is a need to enable this process to take place during field deployment in an interoperable way. M2M service provisioning assumes the existence of an M2M service subscription contracted with the target M2M Service Provider for the target M2M node. Remote provisioning scenarios require the M2M node to be mutually authenticated using pre-existing credentials (e.g. Node-ID and associated credential) with an M2M enrolment function, to securely exchange the provisioning information with the contracted M2M Service Provider. The M2M Service Provisioning takes place between an M2M node (without provisioned CSE) and an M2M Service Provider via an M2M enrolment function. As a result of provisioning, M2M Nodes are provided with necessary credentials and possibly other M2M service related parameters (e.g. CSE-ID, M2M-Sub-ID).

The first step of M2M service provisioning is the security provisioning procedure, by which M2M service provider specific credentials are shared between the M2M node in the field domain and an M2M authentication function in the infrastructure. Authenticated M2M nodes can then be associated with an M2M Service Subscription used to determine their specific authorizations.

The following security provisioning scenarios are supported by the oneM2M architecture:

1) Pre-provisioning:

   - Pre-provisioning includes all forms of out-of band provisioning, e.g. provisioning M2M nodes with M2M subscription information during the manufacturing stage.

2) Remote provisioning:

   - Remote provisioning relies on pre-existing credentials in M2M Nodes (e.g. digital certificates or network access credentials) to provision subscription related parameters through a secure session with an M2M Enrolment Function. This form of provisioning enables M2M nodes already in the field (e.g. operational M2M Nodes) to be provisioned with M2M Service subscription.
   - When supported, remote provisioning procedure shall be implemented as described in the oneM2M security specification [1].
   - Following M2M service provisioning, a CSE associated with the target M2M Service provider in ASN/MN securely stores credentials used for authentication in association with M2M Authentication Function, with an associated lifetime (e.g. corresponding to the duration of the contractual agreement embodied by the M2M service subscription).

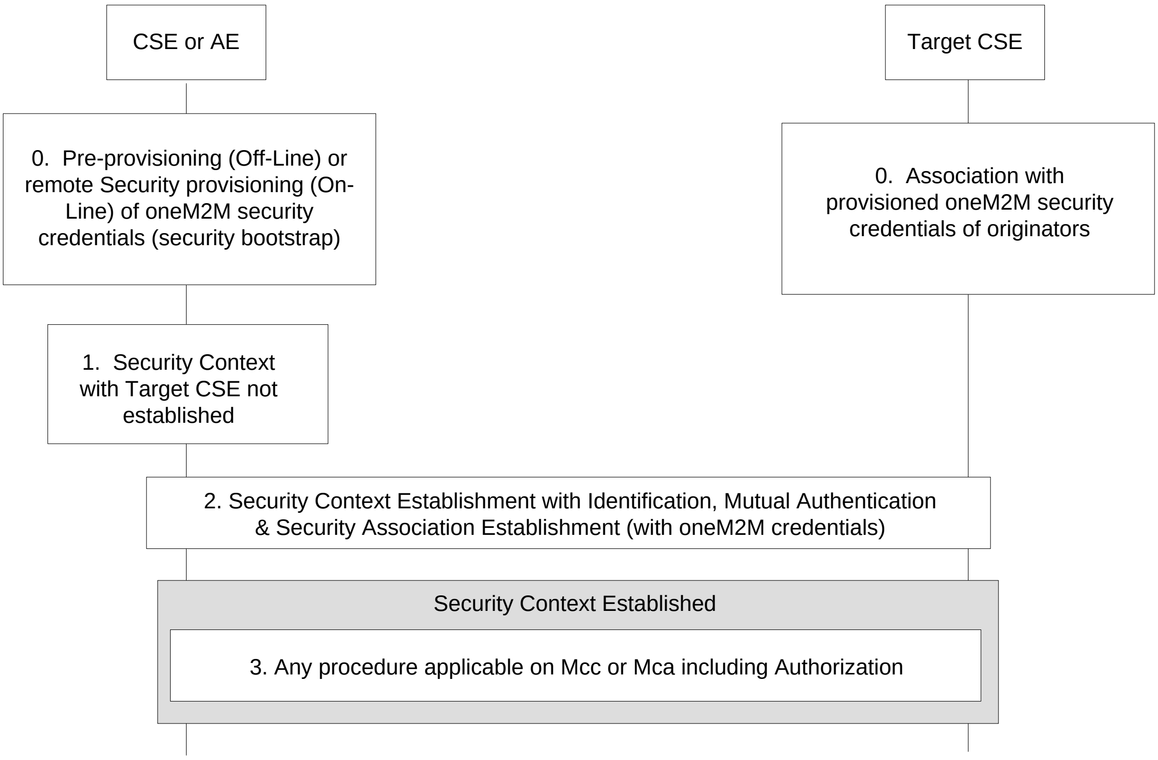
11.2.2 M2M Application Enrolment

This procedure is an optional step that enables the M2M SP and/or M2M application provider to control which applications are allowed to use the M2M services. It assumes that M2M applications obtains or registers credentials to be used for controlling authorization. Each application will then be provisioned with a security credential (M2M Application key) which can be used to grant specific authorization to access an approved list of M2M services. Such authorization takes place between a CSE and an AE.

When supported, the application enrolment procedure shall be implemented as specified in the oneM2M security specification [1].

11.3 M2M Operational Security Procedures

This clause introduces high level procedures that shall be performed before any other procedure on Mcc and Mca can take place.



CSE or AE
Target CSE
0. Pre-provisioning (Off-Line) or remote Security provisioning (On-Line) of oneM2M security credentials (security bootstrap)
0. Association with provisioned oneM2M security credentials of originators
1. Security Context with Target CSE not established
2. Security Context Establishment with Identification, Mutual Authentication & Security Association Establishment (with oneM2M credentials)
Security Context Established
3. Any procedure applicable on Mcc or Mca including Authorization

Figure 11.3-1. High Level Procedures on Mcc or Mca

11.3.1 Identification of CSE and AE

Identification is the process of identifying CSEs and AEs with the associated M2M service subscription to an M2M Authentication Function.

The Identification procedure shall be implemented as specified in the oneM2M security specification [1].

11.3.2 Authentication of CSE and AE

Prior to granting access to M2M services, the credentials resulting from the M2M node and M2M application enrolment procedures shall be used, together with the identities supplied in the identification step, to perform mutual authentication of the entities (AEs or CSEs) with an M2M Authentication Function. Upon mutual authentication, the corresponding entities receive authorization to access the M2M services defined in the M2M Service Subscription.

The Authentication procedure shall be implemented as specified in the oneM2M security specification [1].

11.3.3 M2M Security Association Establishment

The M2M Security Association Establishment procedure is performed to generate a security credential (M2M Connection key) shared between communicating AEs/CSEs, when an AE/CSE on one node initiates communication with an AE/CSE on another node. This procedure is performed after successful identification and mutual authentication of the corresponding M2M entities and derives resulting keys that may be used to provide desired security services to the communicating entities, such as confidentiality and/or integrity of information exchange (these security services may be provided through establishment of a secure channel between the communicating entities or through object based security where only relevant information is encrypted prior to being shared). The lifetime of a security association shall be shorter than the lifetime of the credential used for authentication from which it is derived: It may be valid for the duration of a communication session, or be determined according to the validity period of the protected data. In case of a security association between two AEs, the lifetime of the security association can result from a contractual agreement between the subscribers of the communicating AEs.

The security association establishment procedure shall be implemented as specified in the oneM2M security specification [1].

11.3.4 M2M Authorization Procedure

The M2M authorization procedure controls access to resources and services by CSEs and AEs. This procedure requires that the originator has been identified to an M2M Authentication Function and mutually authenticated and associated with an M2M Service Subscription. Authorization depends on:

- The privileges set by the M2M Service Subscription associated with the originator (e.g. service/role assigned to the originator).

- These privileges are set-up based on the access control policies associated with the accessed resource or service. They condition the allowed operations (e.g. CREATE) based on the originator's privileges and other access control attributes (e.g. contextual attributes such as time or geographic location).

The authorization/access grant involves an Access Decision step to determine what the authenticated CSE or AE can actually access, by evaluating applicable access control policies based on the CSE or AE privileges. Access Decision is described in [1].

The following set of access control policy attributes shall be available for an Access Decision.

- Access control attributes of Originator (e.g. Role, CSE_IDs, AE-IDs, etc.).

- Access control attributes of Environment/Context (e.g. time, day, IP address, etc.).

- Access control attributes of Operations (e.g. Create, Execute, etc.).

The M2M Service Provider/administrator and owner of resources are responsible to establish access control policies that determine by whom, in what context and what operations may be performed upon those resources. If the requesting entity satisfies the owner's access control policy, then the access to the resource is granted.

The authorization procedure involves rerouting of access requests to an M2M authorization function and delivering access tokens valid for specific authorization.

The authorization procedure shall be implemented as specified in the oneM2M security specification [1].

12 Information Recording

12.1 M2M Infrastructure Node (IN) Information Recording

Various informational elements have to be recorded by the M2M infrastructure nodes for a variety of reasons including but not limited to statistics, charging, maintenance, diagnostics, etc.

This clause describes a framework for recording the necessary information by infrastructure nodes.

12.1.1 Information Recording Triggers

Triggers have to be configured in the IN node by the M2M service provider to initiate information recording.

The M2M infrastructure nodes shall be able to initiate recording based on any of the following triggers:

- A request received by the M2M IN over the Mcc reference point.
- A request received by the M2M IN over the Mca reference point.
- A request initiated by the M2M IN over any reference point.
- Timer- based triggers for non- request based information recording. This trigger is used only when the memory size of a container over a period of time is required.

More than one trigger can be simultaneously configured.

The recording triggers may also be configurable, for example, as follows:

- On a per CSE basis, or a group of CSEs for requests originating/arriving from/at the M2M IN.
- On a per AE basis or a group of AEs.
- The default behavior is that no CSEs/AEs are configured.

12.1.2 M2M Recorded Information Elements

12.1.2.1 Unit of Recording

A unit of recording refers to a number of informational elements recorded by the IN and that can be used as a basis for additional post-processing for a specific purpose such as generating Charging Data Records (CDRs), statistics, etc. In that respect, each unit of recording can be thought of as an M2M information record. The actual informational elements that make up a recording unit shall be described later.

For request-based triggers, as defined in clause 12.1.1, the unit of recording shall include a request and its response.

A unit of recording shall be referred to as an M2M Event Record. This shall apply to all recording triggers as defined in clause 12.1.1.

12.1.2.2 Information Elements within an M2M Event Record

The information elements within an M2M event record are defined in table 12.1.2.2-1.

Every M2M event record shall be tagged to depict its content according to the following classification:

- Data related procedures: represent procedures associated with data storage or retrieval from the M2M IN (e.g. Container related procedures).
- Control related procedures: represent all procedures that are not associated with data storage/retrieval from the M2M IN with the exclusion of group and device management related procedures (e.g. subscription procedures, registration).

- Group related procedures: represent procedures that handle groups. The group name may be derived from the target resource in these cases.

- Device Management Procedures.

- Occupancy based trigger for recording the occupancy as described in clause 12.1.1.

Table 12.1.2.2-1: Information Elements within an M2M Event Record

Information Element	For request based triggers Mandatory / optional	For timer based triggers Mandatory / optional	Description
M2M Subscription Identifier	M	M	The M2M subscription ID associated with the request. This is inserted by the IN (see clause 12.1.3)
Application Entity ID	CM (when applicable)	NA	The M2M Application Entity ID if applicable
External ID	CM (when Applicable)	NA	The external ID to communicate over **Mcn** where applicable
Receiver	M	NA	Receiver of an M2M request (can be any M2M Node)
Issuer	M	NA	Issuer of the M2M request (can be any M2M node)
Hosting CSE-ID	O	NA	The hosting CSE-ID for the request in case the receiver is not the host, where applicable
Target ID	M	NA	The target URL for the M2M request if available. Alternatively can be the target resource identifier
Protocol Type	O	NA	E.g. HTTP, CoAP
Request Operation	O	NA	Request Operation as defined in clause 8.1.2
Request Headers size	O	NA	Number of bytes for the headers in the Request, or number bytes of control information
Request Body size	O	NA	Number of bytes of the body transported in the Request if applicable
Response Headers size	O	NA	Number of bytes for the headers in the Response or number bytes of control information
Response Body size	O	NA	Number of bytes of the body transported in the Response if applicable
Response Code	O	NA	
Time Stamp	M	M	The time for the recording the M2M event record
M2M-Event-Record-Tag	M	M	A Tag for the M2M event record for classification purposes. This tag is inserted by the IN and is M2M SP specific
Control Memory Size	O	NA	Storage Memory (in bytes), where applicable, to store control related information associated with the M2M event record(excludes data storage associated with container related operations)
Data Memory Size	O	NA	Storage Memory in Kbytes, where applicable, to store data associated with container related operations
Access Network Identifier	O	O	Identifier of the access network associated with the M2M event record.
Additional Information	O		Vendor specific information
Occupancy	NA	M	Overall size (in Bytes) of the containers generated by a set of AEs identified by the M2M Subscription Identifier
Group Name	CM	NA	Shall be included by the IN in the following cases: - Fanning operation initiated by the M2M IN - A request received by the M2M IN as a result of a fanning operation initiated elsewhere. The group tag shall also be set in this case
maxNrOfMembers	O	NA	Maximum number of members of the group for Create and Update operation.
currentNrOfMembers	O	NA	Current number of members in a group. The request shall be logged and information elements shall be recorded from the request before processing it or sending it out. After obtaining corresponding response, currentNrOfMembers shall be updated with the values from the response.
Subgroup Name	CM	NA	Includes the subgroup member name of a group. It shall be included by the M2M IN in the case when - Fanning operation initiated by the M2M IN and one of the members of the group is a subgroup - A request received by the M2M IN as a result of a fanning operation initiated elsewhere. The group tag shall also be set in this case

The choice for the mandatory elements is motivated by the need to include all M2M identifiers within an M2M event record so that it is possible to support multiple charging scenarios.

For all non-mandatory elements, the M2M IN shall be configurable by the M2M service provider to select any additional desired information to be recorded in addition to the mandatory elements.

12.1.3 Identities Associations in Support of Recorded Information

To enable the M2M IN to record the necessary information, as described above, the following associations shall be maintained by the M2M service provider:

- The CSE-ID (for all M2M nodes in the M2M framework) and the allocated M2M subscription ID.
- The AE-ID   and the allocated M2M subscription ID.

For established associations, as described above, the M2M IN shall derive the appropriate M2M subscription ID for insertion in the M2M record event.

12.2 Offline Charging

12.2.1 Architecture

Figure 12.2.1-1 depicts the charging architecture. Charging information, in the form of charging data records (CDRs), shall be derived from recorded information, and transferred to a Charging Server. As such, it is essential that all information required for charging shall be first selected for recording. There shall be a 1 to 1 mapping between a M2M Event Record and a CDR.

The Charging Function (CHF included within the SCA CSF) embedded within the M2M IN is responsible for interaction with the Charging Server using the Mch reference point.

Billing aspects are out of scope.

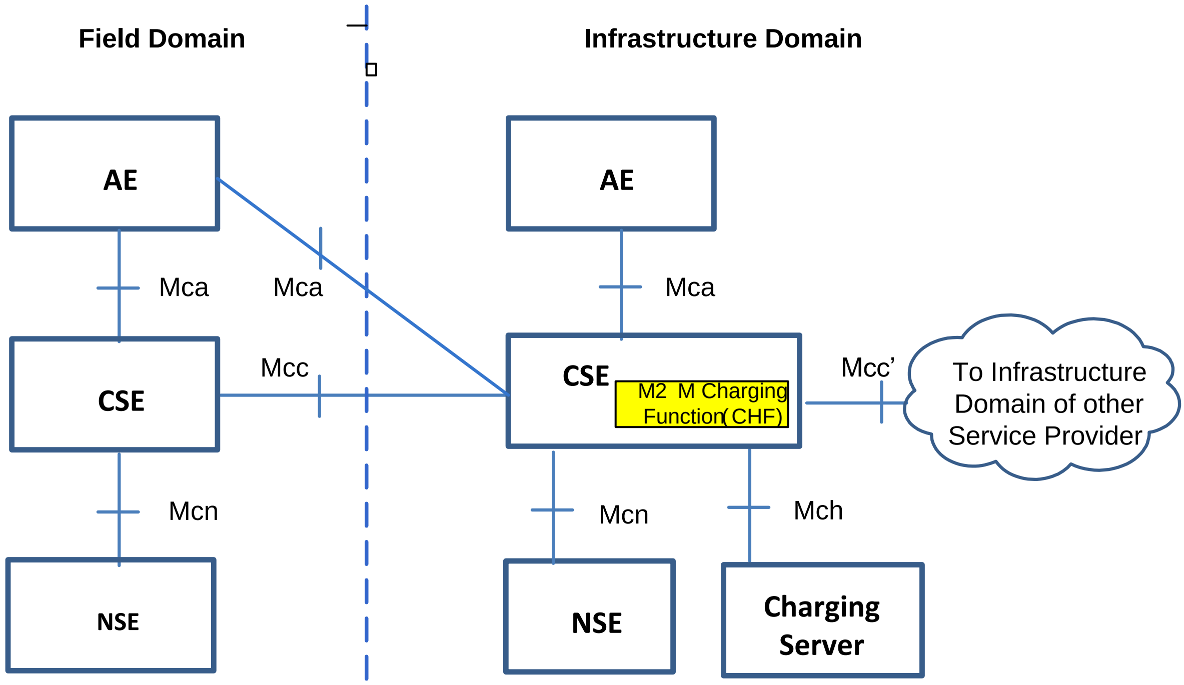


Figure 12.2.1-1: Offline Charging Architecture

[12.2.1.a] Editors Note: Update Figure 5.2.1-1 to include this new interface.

[12.2.1.b] Editor's Note: Once this picture is moved to clause 5.5; need to remove the picture from here.

12.2.2 Filtering of Recorded Information for Offline Charging

Recorded information is the basis for offline charging. To fulfil the needs of different billing systems not all recorded information is required in all cases. Hence, the M2M Charging Function shall be configurable to only select the desired information from the recorded information for transfer to the Charging Server. This configuration shall support selecting the desired information based on the following capabilities:

- On a per CSE basis, or a group of CSEs, for requests originating/arriving from/at the IN. This applies to all M2M nodes within the M2M framework.
- On a per AE basis or a group of AEs.
- The default behavior is that no CSEs/AEs are configured.

The charging function shall ensure that information selected for transfer to the charging server has also been selected for recording before a configuration is deemed acceptable for execution.

12.2.3 Examples of Charging Scenarios

Charging scenarios refer to scenarios for which an M2M entity can be billed if the scenario is deemed billable by the M2M service provider. Some charging scenarios may require single CDR. Other scenarios may require multiple CDRs, and suitable correlation information shall have to be identified to select the CDRs for the charging scenario in this case.

The following clause lists some potential charging scenarios as examples only. Each scenario shall require the appropriate configuration of the CHF, and for that matter the M2M recording functions, to ensure that all pertinent data is available.

12.2.3.1 Example Charging Scenario 1 - Data Storage Resource Consumption

In this scenario, the M2M entity that stores application data, using container procedures for that purpose, will be billed, for storage resources within the M2M IN, until such time as the resources are deleted. This scenario will require correlation between multiple CDRs to identify the entity that stored the data, the entity that deleted the same data, and the duration and amount of storage.

12.2.3.2 Example Charging Scenario 2 - Data transfer

In this scenario, the M2M entity that retrieves/stores container data will be billed for the amount of transferred data.

12.2.3.3 Example Charging Scenario 3 - Connectivity

This scenario is relevant for an M2M entity that contacts the M2M IN frequently to transfer small amounts of data for storage. In this scenario, the M2M entity will be charged for the connectivity as opposed to the stored amount of data. The same applies to an M2M entity that also contacts frequently the M2M IN to retrieve stored data.

12.2.4 Correlation between charging information of M2M IN and charging in access network

[12.2.4.a] Editor's Note: <<FFS>>

12.2.5 Definition of Charging Information

Charging information in the form of CDR is essentially a subset of the information elements within the M2M event records recorded by the M2M IN for transmission over the Mch reference point.

12.2.5.1 Triggers for Charging Information

The charging function within the M2M IN shall initiate transmission of CDRs if configured for that purpose in accordance with clause 12.2.2.

12.2.5.2 Charging Messages over Mch Reference Point

The Mch shall be used in case the CDRs are to be transferred to an external Charging Server. It is assumed that the Mch is equivalent to the Rf reference point as defined in [i.18] and [i.19].

Hence, every CDR shall be transferred in a single message, namely Accounting-Request and that elicits a response, namely Accounting-Answer.

The following table describes the use of these messages for offline charging.

Table 12.2.5.2-1: Offline charging messages reference table

Request-Name	Source	Destination	Abbreviation
Accounting-Request	M2M IN	Charging Server	ACR
Accounting-Answer	Charging Server	M2M IN	ACA

12.2.5.3 Structure of the Accounting Message Formats

12.2.5.3.1 Accounting-Request Message

Table 12.2.5.3.1-1 illustrates the basic structure of an ACR message generated from the M2M IN for offline charging in accordance with [i.18], [i.19], [i.10] and [i.13].

Table 12.2.5.3.1-1: Accounting-Request (ACR) message contents

Informational Element	Category	Description
Session-Id	M	This field identifies the operation session. The usage of this field is left to the M2M SP.
Origin-Host	M	This field contains the identification of the source point of the operation and the realm of the operation originator
Origin-Realm	M	This field contains the realm of the operation originator.
Destination-Realm	M	This field contains the realm of the operator domain. The realm will be addressed with the domain address of the corresponding public URI.
Accounting-Record-Type	M	This field defines the transfer type: This field shall always set to event based charging.
Accounting-Record-Number	M	This field contains the sequence number of the transferred messages.
Acct-Application-Id	O_C	Advertises support for accounting for M2M.
Event-Timestamp	O	Defines the time when the event occurred.
Proxy-Info	O_C	Includes host information about a proxy that added information during routing of the message.
Route-Record	O_C	This field contains an identifier inserted by a relaying or proxying charging node to identify the node it received the message from.
Service-Context-Id	M	This field identifies the M2M domain.
Service-Information	M	This parameter is set to "M2M service-information".
M2M Subscription-Id	M	Identifies the M2M subscription ID.
M2M Information	M	This parameter holds the M2M informational element specified in table 11.1 <provide reference to correct table> with the exception of the M2M subscription ID.
Proprietaryinformation	O	This is for proprietary information.
O_C		This is a parameter that, if provisioned by the service provider to be present, shall be included in the CDRs when the required conditions are met. In other words, an O_C parameter that is configured to be present is a conditional parameter.

[12.2.5.3.1.b] Editors Note: The inclusion of additional information is FFS

12.2.5.3.2 Accounting-Answer Message

The following table illustrates the basic structure of an ACA message generated by the charging server as a response to an ACR message.

Table 12.2.5.3.2-1: Accounting-Answer (ACA) message contents

Information element	Category	Description
Session-Id	M	Same as table 12.2.5.3.1-1
Origin-Host	M	Same as table 12.2.5.3.1-1
Origin-Realm	M	Same as table 12.2.5.3.1-1
Destination-Realm	M	Same as table 12.2.5.3.1-1
Accounting-Record-Type	M	Same as table 12.2.5.3.1-1
Accounting-Record-Number	M	Same as table 12.2.5.3.1-1
Acct-Application-Id	O_C	Same as table 12.2.5.3.1-1
Error-Reporting-Host	O_C	Included only if the host that inserted the error is different from the Origin-Host
Event-Timestamp	O	Same as table 12.2.5.3.1-1
Proxy-Info	O_C	Same as table 12.2.5.3.1-1
Proprietary Information	O	Same as table 12.3.5.3.1-1
O_C	This is a parameter that, if provisioned by the operator to be present, shall be included in the CDRs when the required conditions are met. In other words, an O_C parameter that is configured to be present is a conditional parameter.	

Annex A (informative): Mapping of Requirements with CSFs

Table A-1 illustrates the mapping of the Requirements specified in TS-0002 [i.1] with the CSFs specified in this document.

Table A-1: Mapping of Requirements to CSFs

CSF Name	Supported Sub-Functions	Associated Requirements	Notes
Addressing and Identification (AID)	• Management of identifiers	• OSR-026 • OSR-023 • OSR-024 • OSR-025	Overlap w/: DSC for OSR-023, OSR-024, and OSR-025
Communication Management/Delivery Handling (CMDH)	• Providing communications with other CSE's, AE's, and NSE's • Communications management: best effort • Communications policy management • Underlying Network connectivity management • Communications management: data store and forward • Ability to trigger off-line device	• OSR-001 • OSR-002 • OSR-005 • OSR-006 • OSR-008 • OSR-009 • OSR-012 • OSR-013 • OSR-014 • OSR-015 • OSR-018 • OSR-019 • OSR-021 • OSR-027 • OSR-032 • OSR-035 • OSR-038 • OSR-039 • OSR-040 • OSR-048 • OSR-049 • OSR-050 • OSR-053 • OSR-062 • OSR-063 • OSR-064 • OSR-065 • OSR-066 • OSR-067 • OSR-068 • CRPR-001 • CRPR-002 • CRPR-003 • MGR-016	Overlap w/: DMR for OSR-001, OSR-009, OSR-021, OSR-032 SSM for OSR-009 LOC for OSR-006 GMG for OSR-006 NSSE for OSR-006, OSR-027 SSM for OSR-009
Data Management and Repository (DMR)	• Data storage and management • Semantic support • Data aggregation • Data analytics • Device data backup and recovery	• OSR-001 • OSR-007 • OSR-009 • OSR-016 • OSR-020 • OSR-021 • OSR-032 • OSR-034 • OSR-036 • OSR-058 • SMR-006 • SER-015	Overlap w/: CMDH for OSR-001, OSR-009, OSR-021, OSR-032 SUB for OSR-016 GMG for OSR-020 .

CSF Name	Supported Sub-Functions	Associated Requirements	Notes
Device Management (DMG)	• Configuration Management • Diagnostics and Monitoring • Firmware management • Software management • Device Area Network topology management	• OSR-017 • OSR-069 • OSR-070 • OSR-071 • OPR-001 • OPR-002 • OPR-003 • MGR-001 • MGR-003 • MGR-004 • MGR-006 • MGR-007 • MGR-008 • MGR-009 • MGR-011 • MGR-012 • MGR-013 • MGR-014 • MGR-015 • MGR-019 • MGR-020 • MGR-021 • SER-013 • SER-014	Overlaps w/: GMG for OSR-017 SEC for SER-013
Discovery (DIS)	• Discover resource • Local discovery (within CSE) • Directed remote discovery	• OSR-023 • OSR-024 • OSR-025 • OSR-059 • OSR-060 • OSR-061 • MGR-002 • SMR-004	Overlaps w/: AID for OSR-023, OSR-024, OSR-025
Group Management (GMG)	• Management of a group and its membership • CRUD • Use Underlying Network group capabilities • Bulk operations • Access control	• OSR-006 • OSR-017 • OSR-020 • OSR-029 • OSR-030 • OSR-031 • OSR-037 • OSR-047 • MGR-005	Overlaps w/: CMDH for OSR-006 LOC for OSR-006 GMG for OSR-006 NSSE for OSR-006, OSR-037 DMR for OSR-020 DMG for OSR-017
Location (LOC)	• Location management • Network-provided • GPS-provided • Confidentiality enforcement as it relates to location	• OSR-006 • OSR-051 • OSR-052 • SER-026	
Network Service Exposure /Service execution and triggering (NSSE)	• Access Underlying Network service • Location • Device triggering • Small data • Policy and charging • Support multiple Underlying Network functions	• OSR-006 • OSR-011 • OSR-027 • OSR-037 • OSR-054 • OSR-055 • OSR-056 • MGR-017 • MGR-018 • OPR-004 • OPR-005 • OPR-006	Overlaps w/: CMDH for OSR-027 GMG for OSR-006, OSR-037 LOC for OSR-006
Registration (REG)	• CSE registration • Application registration • Device registration • ID correlation	• MGR-010	Overlaps w/: SEC

CSF Name	Supported Sub-Functions	Associated Requirements	Notes
Security (SEC)	• Sensitive Data Handling • Secure storage • Secure execution • Independent environments • Security administration • Pre-provisioning • Dynamic bootstrap • Network bootstrap • Security association • Link level • Object level • Authorization and access • Identity protection	• SER-001 • SER-002 • SER-003 • SER-004 • SER-005 • SER-006 • SER-007 • SER-008 • SER-009 • SER-010 • SER-011 • SER-012 • SER-013 • SER-016 • SER-017 • SER-018 • SER-019 • SER-020 • SER-021 • SER-022 • SER-023 • SER-024 • SER-025 • MGR-010	Overlap w: DMG for SER-013 REG for MGR-010 SSM for SER-007
Service Charging and Accounting (SCA)	• Charging enablers • Sending charging information to charging server • Subscription-based charging • Event-based charging • Session-based charging • Service-based charging • Correlation with Underlying Network • Charging management • Offline charging • Online charging	• CHG-001, • CHG-002a, • CHG-002b, • CHG-003, • CHG-004, • CHG-005	
Service Session Management (SSM)	• Service Session Management (CSE to CSE, AE to CSE, and AE to AE) • Session persistence over link outage • Session context handling • Assignment of session ID • Session routing • Multi-hop session management • Session policy management	• OSR-003 • OSR-004 • OSR-009 • OSR-045 • SER-007	Overlap w/: CMDH and DMR for OSR-009 SEC for SER-007
Subscription/Notification Support (SUB)	• Subscribe (CSE, AE) • Local • Remote • Subscription to a group • Notification • Synchronous • Asynchronous	• OSR-010 • OSR-016 • OSR-033	Overlaps w/: DMR for OSR-016

Annex B (informative): oneM2M System and 3GPP MTC Release-11 Underlying Network Interworking

B.1 3GPP MTC Release-11 Underlying Network Introduction

In order to provide M2M services, interworking between oneM2M System and the 3GPP Underlying Network is required. This entails the following aspects:

- IN-CSE initiated connectivity establishment between the oneM2M System and the 3GPP Underlying Network from IN-CSE initiated scheme and device (i.e. ASN-CSE) initiated scheme;
- M2M device (e.g., ASN/MN-CSE) initiated connectivity establishment between the oneM2M System and the 3GPP Underlying Network;
- Mapping of oneM2M and 3GPP specific Identifiers to establish connectivity between specific entities.

This clause provides system level information on the above aspects specifically related to the interworking i.e. connectivity between the oneM2M System and the 3GPP Underlying Network.

B.2 3GPP Release-11 MTC Functionality

Interworking with oneM2M Release-1 is based on 3GPP Release-11 specifications. The relevant 3GPP Release-11 specification references are as follows:

- 3GPP TS 23 682[i.17]: Architecture Enhancements to facilitate Communication with Packet Data Networks and Applications;
- 3GPP TS 23 401 [i.22]: GPRS Enhancements for E-UTRAN Access;
- 3GPP TS 23 402 [i.23]: Architecture Enhancements for non-3GPP Accesses;
- 3GPP TS 23 060 [i.24]: General Packet Radio Service (GPRS) Service Description;
- 3GPP TS 22.368 [i.25]: Service requirements for Machine Type Communications (MTC); Stage 1.

In annex A of TS 123 682 [i.17] the following MTC deployment scenarios are depicted.

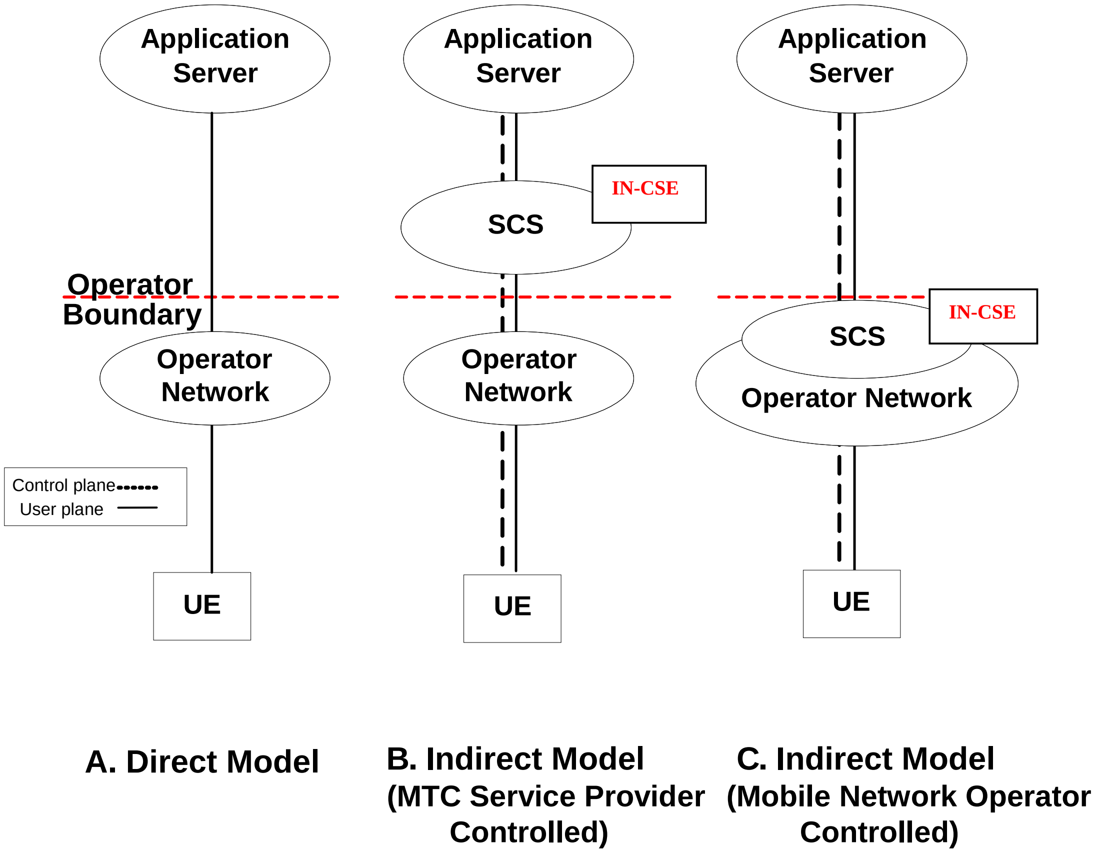


Figure B.2-1: MTC deployment scenarios for Direct and Indirect model

The focus of this annex is on deployment scenario B (Indirect Model), where M2M Services Capability Server is outside the 3GPP operator domain. The indirect model, scenario C in figure B.2-1, where M2M Service Capability Server is inside the 3GPP operator domain is also not ruled out.

Hybrid model which is a combination of above scenario B and C is also mentioned in TS23 682 [i.17] which may also be supported.

Taking 3GPP Release-11 MTC network as the Underlying Network, oneM2M IN-CSE is considered as equivalent to or part of the Services Capability Server (SCS), oneM2M ASN/MN is considered equivalent to a UE, as captured in 3GPP MTC architecture (TS 23 682 [i.17]) shown below.

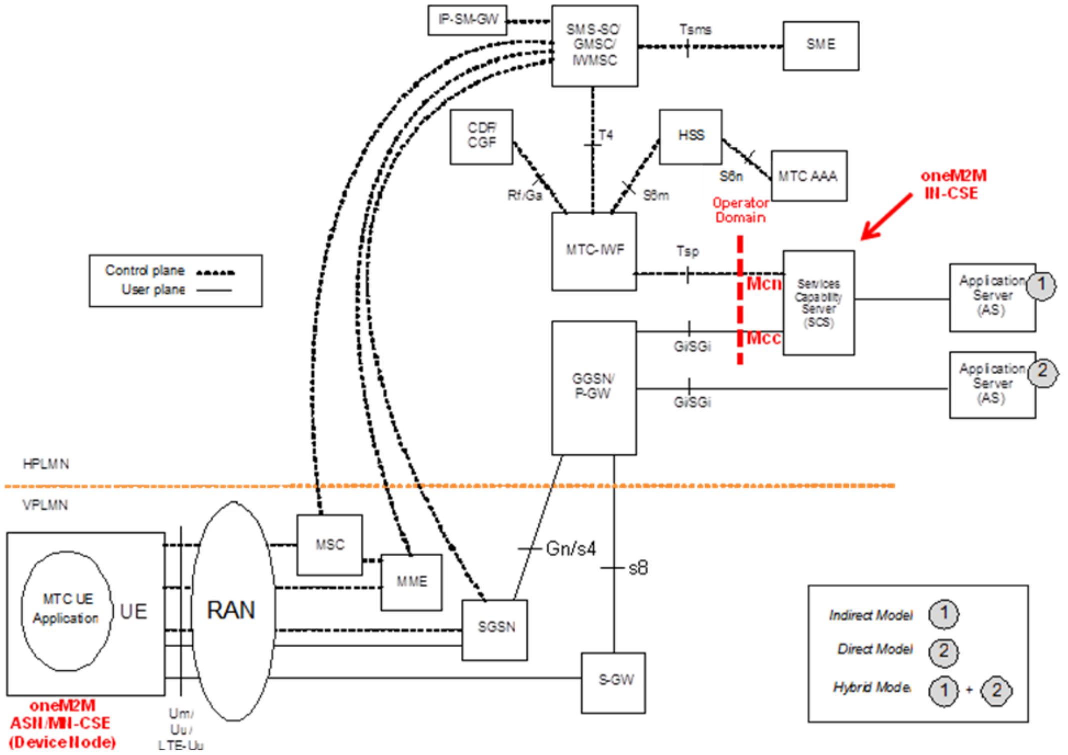


Figure B.2-2: 3GPP Architecture for Machine-Type Communication

IN-CSE can be inside or outside the 3GPP operator domain. The IN-CSE interacts with the 3GPP Underlying Network via MTC-IWF and/or GGSN/P-GW. This requires mapping of oneM2M reference points (Mcn, Mcc) and 3GPP reference points (Tsp, Gi/SGi) respectively, along with the mapping of the identifiers in the two systems.

B.3 ASN/MN-CSE initiated connectivity establishment

It is assumed that there is no connectivity previously established, i.e. no association between the ASN/MN-CSE (device Node) and the serving IN-CSE exists. When the ASN/MN-CSE needs to send data to the serving IN-CSE it first discovers the serving IN-CSE, which is located in a packet data network, and establishes connection. Two methods can be used, as follows:

- Use of DHCP and DNS
- Pre-configuration

B.3.1 Use of DHCP and DNS

The ASN/MN-CSE requests the DNS server address from the DHCP server followed by requesting the serving IN-CSE IP address from the DNS server.

NOTE 1: How a non-CSE capable M2M device (e.g., ADN) interworks with the 3GPP Underlying Network is not specified in this release of the document.

B.3.2 Pre-configuration

The ASN/MN-CSE is preconfigured with the fully qualified domain name (FQDN) of the serving IN-CSE or the IP address of the serving IN-CSE. If the FQDN is known, DNS resolution is used to obtain the IP address.

B.4 Serving IN-CSE initiated connectivity establishment

It is assumed that there is no connectivity previously established between the ASN/MN-CSE and the serving IN-CSE. When the serving IN-CSE needs to contact the ASN/MN-CSE to send data or request data, connectivity between them is established. This connectivity is triggered by the IN-CSE.

NOTE 2: How the IN-CSE triggers a non-CSE capable M2M device (e.g., ADN) within the 3GPP Underlying Network is not specified in this release of the document.B.5 Connectivity between oneM2M Service Layer and 3GPP Underlying Network

ASN/MN-CSE communicates with the serving IN-CSE after completion of the Underlying Network bearer establishment and discovery of the serving IN-CSE. Data can then traverse between CSEs over the IP connection in the Underling Network over 3GPP Gi/SGi interface. In addition, the signalling connectivity between the two CSEs is also realized. The following figure depicts the connectivity between the ASN/MN-CSE and the IN-CSE.

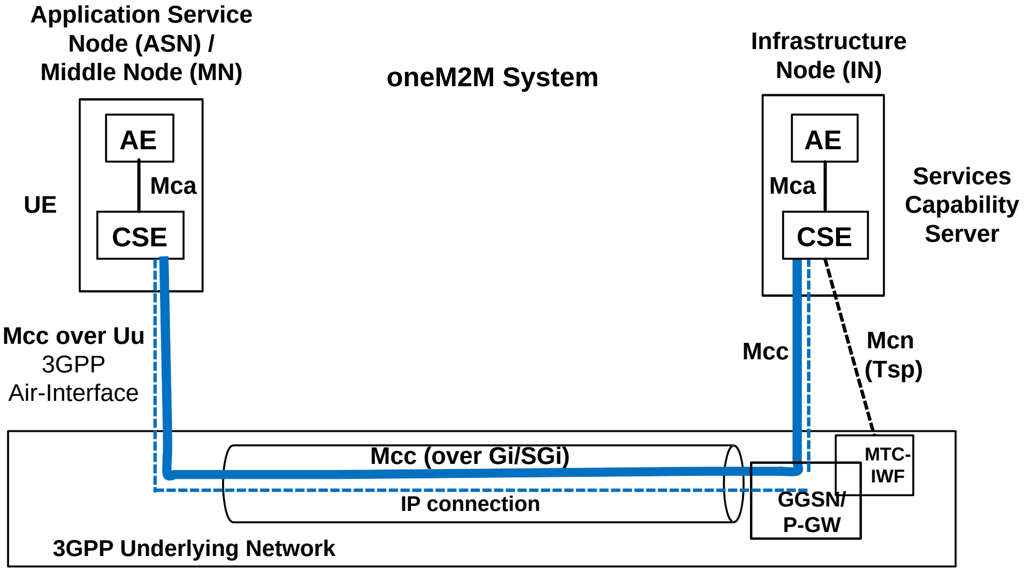


Figure B.5-1: Connectivity Establishment between ASN/MN-CSE and IN-CSE

B.6 Connectivity Establishment Procedures

B.6.1 General

When data is to be exchanged between the ASN/MN-CSE and the IN-CSE, connectivity between them needs to be established. The need for this connectivity can arise for two reasons:

1) ASN/MN-CSE initiated: When the ASN/MN-CSE needs to send/receive data to/from the IN-CSE; or

2) IN-CSE initiated: When the IN-CSE needs to send/receive data to/from the ASN-CSE.

Connectivity establishment procedures in this clause are example illustrations and do not exclude other realizations.

This is a draft oneM2M document and should not be relied upon; the final version, if any, will be made available by oneM2M Partners Type 1.

B.6.1.1 ASN/MN-CSE Initiated Connectivity Establishment Procedure

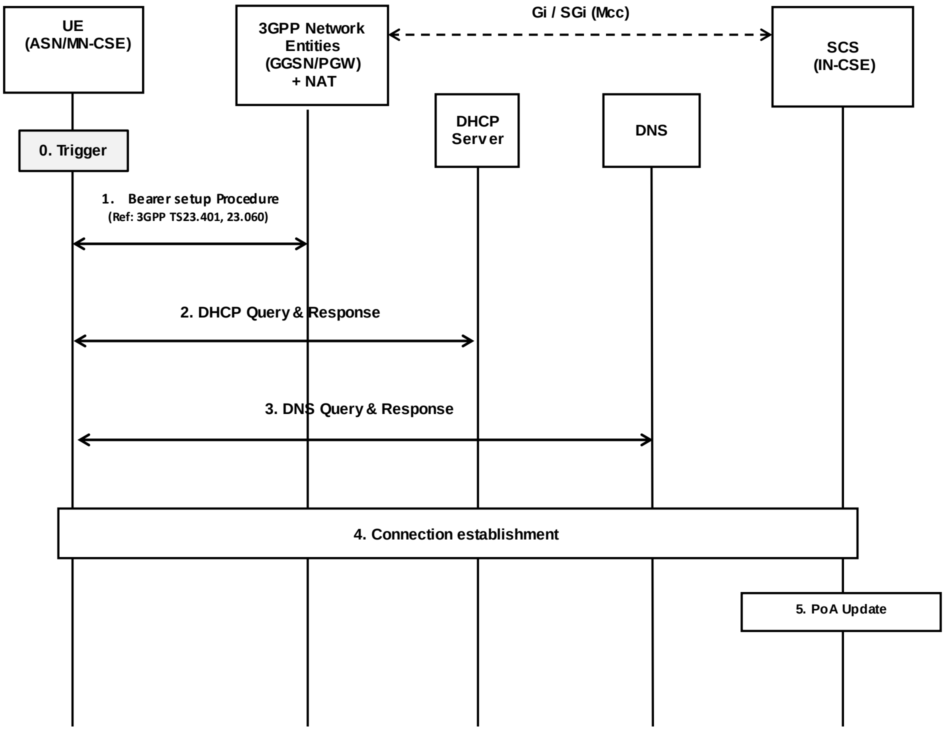


Figure B.6.1.1-1: ASN/MN-CSE initiated connectivity establishment

Step-0: Trigger

Subsequent procedures are triggered either when the ASN/MN-CSE powers on or resulting from Device Triggering mentioned in clause B.6.1.2.

Step-1: Bearer Setup Procedure

Establish a 3GPP bearer(s) if not already available by using the procedures available in the 3GPP network.

Step-2: DHCP Query & Response

The ASN/MN-CSE sends a query to a DHCP server to find a particular DNS server IP address. The DHCP server responds with the IP address of a corresponding DNS server. Additionally, it is also possible to include one or a list of domain names, i.e. FQDNs of target IN-CSEs.

Step-3: DNS Query & Response

The ASN/MN-CSE performs a DNS query to retrieve the IN-CSE(s) IP addresses from which one is selected. If the response does not contain the IP addresses, an additional DNS query is needed to resolve a Fully Qualified Domain Name (FQDN) of the serving IN-CSE to an IP address.

Step-4: Connection Establishment

After reception of domain name and IP address of an IN-CSE, the ASN/MN-CSE can initiate communication towards the IN-CSE via the IP connection. The IN-CSE at this time shall be informed which Trigger Recipient ID of the ASN/MN-CSE to use for establishing communication.

Step-5: CSE-PoA Update

Once the M2M Service Connection (Mcc) is established, in the IN-CSE the CSE-PoA of the ASN-CSE/MN-CSE shall be updated with the new established IP address.

The IN-CSE holds the state information and needs to be informed when the connection is closed.

B.6.1.2 IN-CSE initiated connectivity establishment procedure over Tsp

Connection Establishment between IN-CSE and ASN/MN-CSE

Whenever the IN-CSE requires to establish a connection towards another entity (e.g., ASN/MN-CSE), Device Triggering procedure over the Tsp interface as described in TS 23 682 [i.17] shall be used.

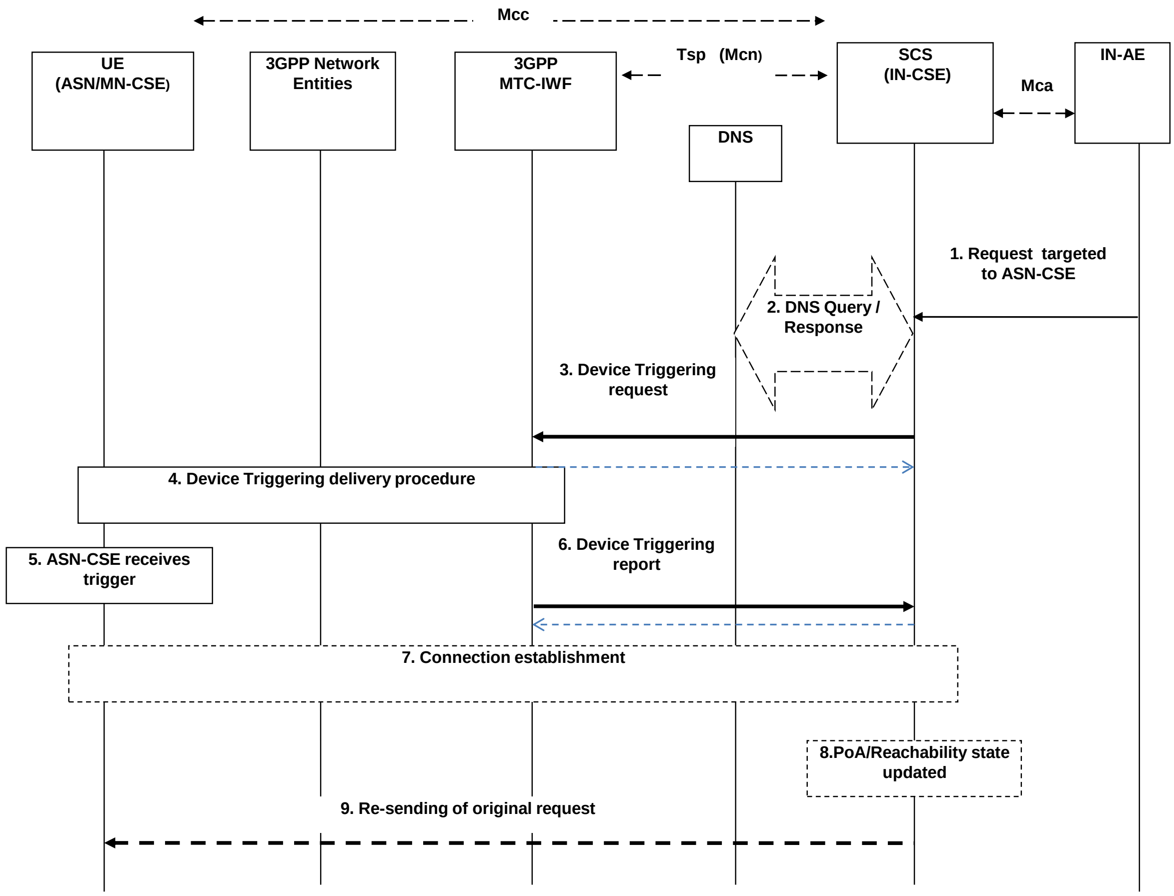


Figure B.6.1.2-1: IN-CSE initiated connectivity establishment

Pre-condition

The CSE which is the target of the device triggering has to be registered with the IN-CSE. The IN-CSE checks the state information of the target device. Some of this state information is the result of a previous connection establishment or triggering requests, such as the case of power-off, dormant and/or connected state. The IN-CSE decides its next action, e.g., if it needs to start device triggering or to report to IN-AE about the inability to perform the request.

The CSE-PoA for the ASN/MN-CSE either already contain an IP address which is not valid anymore or no IP address at all, or FQDN does not resolve to a valid IP address. This is a pre-requisite for performing the device triggering procedure.

[optional] Step 1: Request targeted to ASN/MN-CSE

The IN-AE requests to perform one of the CRUD operation on a resource residing on the ASN/MN-CSE, the request is sent via the Mca reference point to the IN-CSE. The request from IN-AE includes the address (URI) of the target resource.

Step 2: DNS Query / Response

The IN-CSE determines the need to trigger the ASN/MN-CSE.

If the IN-CSE has no contact details for a contact MTC-IWF, it may determine the IP address(es)/port(s) of the MTC-IWF by performing a DNS query using the M2M-Ext-ID (M2M External Identifier) assigned to the target ASN/MN-CSE, or using a locally configured MTC-IWF identifier.

Step-3: Device Triggering Request

The IN-CSE buffers the original request information and sends the Device Trigger Request message that contains information as specified in 3GPP TS 23.682 [i.17]. Such information includes:

- M2M-Ext-ID or MSISDN;
- SCS-Identifier, (is set to the IN-CSE ID);
- Trigger reference number (used to correlate the request with the response);
- Validity period, (which indicates how long the request is valid);
- Priority (this field allows to set the priority on or off);
- Application Port ID, (is set to the ASN/MN-CSE Trigger-Recipient-ID since it is the triggering application addressed in the device from 3GPP point of view);
- Trigger payload, (optional information can be set to the payload).

NOTE 3: In case that the Device Triggering request is for an M2M Service Connection setup request as in the present flow, it is assumed that when the target CSE (i.e. ASN/MN-CSE) is woken up on receiving the trigger it initiates connection establishment with the IN-CSE with whichit is registered. The information of the IN-CSE may be pre-stored in the target CSE (i.e. ASN/MN-CSE). Therefore it is assumed that the trigger payload does not include the optional information and the target CSE is registered to only one IN-CSE. How to use the optional part of the trigger payload is described as below.

Acknowledge

Once, 3GPP-MTC-IWF receives the Trigger Request, it asks the HSS to determine if the IN-CSE is authorized to perform the triggering to the target CSE (i.e. ASN/MN-CSE) and the HSS resolves the M2M-Ext-ID to IMSI (or MSISDN). Then the 3GPP MTC-IWF acknowledges to the IN-CSE with the confirmation of receiving Device Triggering Request.

Step-4: Device Triggering delivery procedure

The MTC-IWF initiates the T4 trigger delivery procedure according to the TS 123 682 [i.17], based on the information received from HSS and local policy.

NOTE 4: 3GPP Network Entities (e.g. SMS-SC) can select appropriate device triggering mechanism (e.g. SMS based or SIP based via IP-SM-GW) according to the device capabilities.

Step 5: ASN/MN-CSE receives the trigger

As a result of the device triggering procedure the addressed ASN/MN-CSE is initiated/ In this case of the flow the ASN/MN-CSE starts according to the received Application Port ID by the UE.

NOTE 4: In case the Device Trigger contains the optional part of the trigger payload, it is assumed that such trigger payload e is forwarded to the application inside the ASN/MN-CSE that is started as a result of device trigger.

Step 6: Device Triggering report

Request:

The MTC-IWF sends the Device Trigger Report message (containing the M2M-Ext-ID or MSISDN and trigger reference number) to the IN-CSE with a cause value indicating whether the trigger delivery succeeded or failed and the reason for the failure.

Acknowledge:

IN-CSE acknowledges to the MTC-IWF with the conformation of the received Device Triggering Report

[optional] Step 7: Connection establishment procedure

The ASN/MN-CSE performs the Connection establishment procedure as described in clause B.6.1.1 and oneM2M TS-0003 [i.3] for Secure Connection establishment.

As a result of this procedure the initial request over the reference point Mcc can be executed.

[optional] Step 8: CSE-PoA/Reachability state updated

Once the connection over Mcc is established, the PoA of the ASN/MN-CSE shall be updated at the IN-CSE with the new established IP address and the IN-CSE holds the reachability state of the ASN/MN-CSE.

[optional] Step 9: Re-sending of original request

As a result of step 7, the communication is established and now the initial request with the information stored in the buffer of the IN-CSE at Step 3 can be re-issued over the reference point Mcc.

In the flow presented above not all parameters allowed in the Device Triggering Request message from 3GPP Tsp interface are used. Optionally the following cases are allowed:

By providing a payload which may contain:

Either actual content information (as permitted by the limitation of the payload parameter). For example;

- It could contain a resource (or attribute) identifier (as expressed inside the ASN/MN-CSE) and the actual content for the resource (or attribute) of any of the resources stored in the ASN/MN-CSE.

- Or any other instructions for initiating a specific procedure. For example, to execute a command.

- Or it could contain of the URI of an entity outside the oneM2M domain where the target ASN/MN-CSE should connect to. If a URI is provided, the steps 7, 8 and 9 of the previous flow are performed since the connection establishment is not performed between two oneM2M entities. How the actual setup with an entity outside the oneM2M domain is performed it is outside the scope of this specification.

Annex C (informative): Interworking between oneM2M System and 3GPP2 Underlying Networks

C.1 General Concepts

Interworking between oneM2M System and 3GPP2 Underlying Networks is based on 3GPP2 X.P0068 specification [i.20].

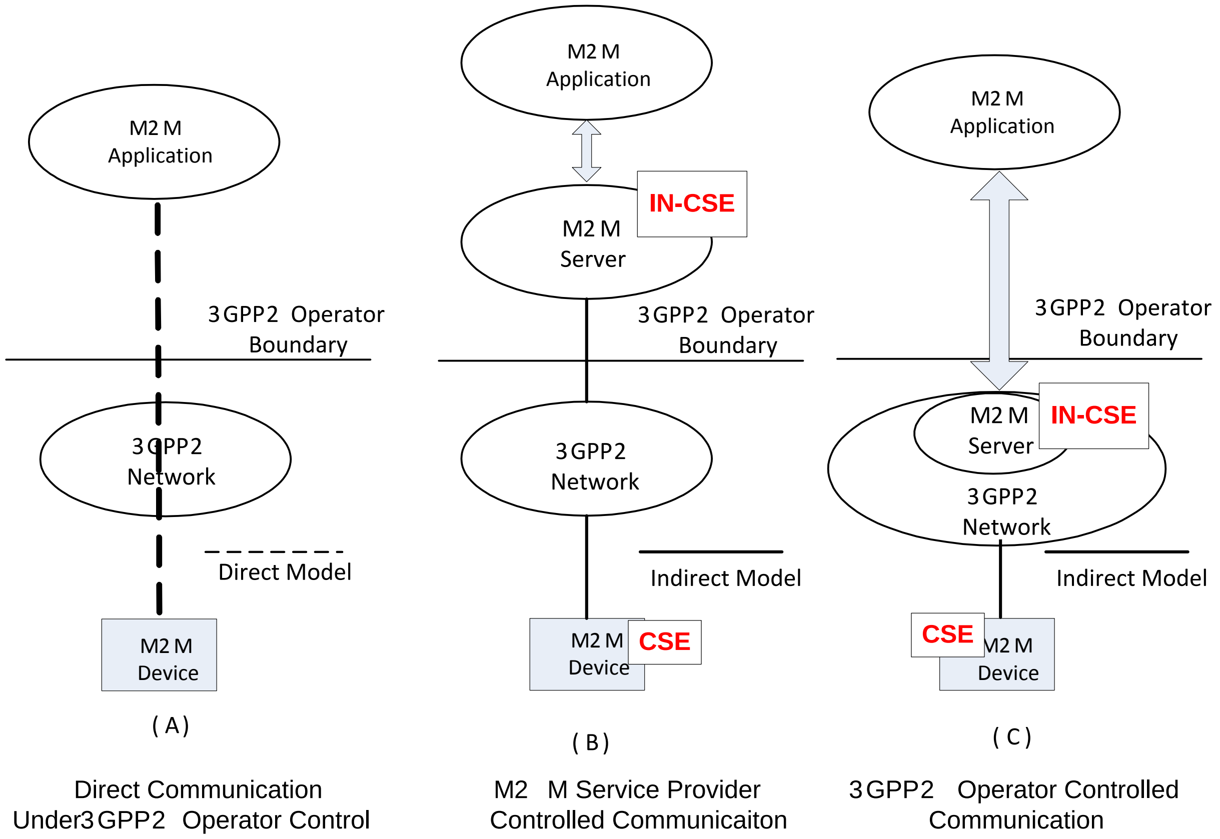
In order to provide M2M services, interworking between oneM2M System and the 3GPP2 Underlying Network is required. M2M Applications (AEs) in the M2M UEs (M2M Nodes such as the ASNs and MNs) and the M2M Applications in the external network (Infrastructure Domain) use services provided by the 3GPP2 Underlying Network, and optionally the services provided by an M2M Server (IN-CSE). The 3GPP2 Underlying Network provides transport and communication services, including 3GPP2 bearer services, IMS and SMS.

3GPP2 Underlying Network supports several interworking models, such as the following:

- Direct Model - Direct Communication provided by the 3GPP2 Network Operator:
  - The M2M Applications in the external network connect directly to the M2M Applications in the UEs used for M2M via the 3GPP2 Underlying Network without the use of any M2M Server.
- Indirect Model - M2M Service Provider controlled communication:
  - Uses an M2M Server that is an entity outside the 3GPP2 Underlying Network operator domain for enabling communications between the Applications in the external network and at the UEs used for M2M. Tsp interface is an external interface that the third party M2M Server supports with the entities that are within the 3GPP2 Underlying Network domain.
- Indirect Model - 3GPP2 Operator controlled communication:
  - Uses an M2M Server that is an entity inside the 3GPP2 Underlying Network operator domain for enabling communications between the Applications in the external network and at the UEs used for M2M. Tsp interface is an internal interface that the 3GPP2 Underlying Network operator controlled M2M Server supports with other entities within the 3GPP2 Underlying Network domain.
- Hybrid Model:
  - Direct and Indirect models are used simultaneously in the hybrid model i.e. performing Control Plane signalling using the Indirect Model and connecting the M2M Applications in the external network and at the UEs used for M2M over User Plane using the Direct Model.

C.2 M2M Communication Models

In the indirect and hybrid models, the deployment of an M2M Server (IN-CSE) may be inside or outside the 3GPP2 Underlying Network operator domain as illustrated in figures C.2-1 and C.2-2. When the M2M Server is part of the 3GPP2 Underlying Network operator domain (figures C.2-1(C) and C.2-2), the M2M Server is considered a 3GPP2 Underlying Network operator internal network function, is operator controlled, and may provide operator value-added services. In this case, security and privacy protection for communication between the M2M-IWF and the M2M Server (IN-CSE) is optional. When the M2M Server is deployed outside the 3GPP2 Underlying Network operator domain (figures C.2-1(B) and C.2-2), the M2M Server is M2M Service Provider controlled. In this case, security and privacy protection for communication between the M2M-IWF and the M2M Server (IN-CSE) is needed. In the direct model (figure C.2-1(A)), there is no external or internal M2M Server in the communication path.



Direct Communication Under3GPP2 Operator Control

M2 M Service Provider Controlled Communicaiton

3GPP2 Operator Controlled Communication

Figure C.2-1: M2M Communication Models

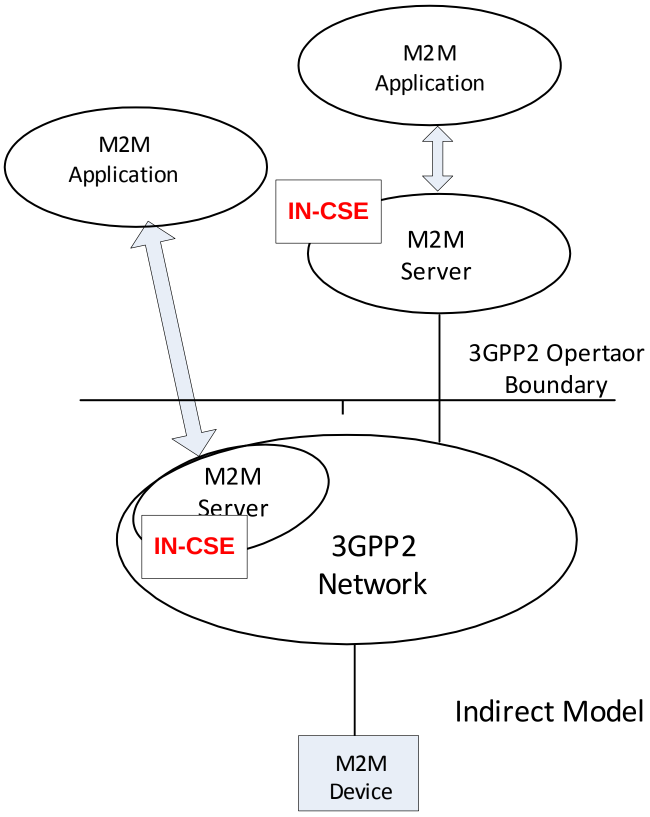


Figure C.2-2: Multiple M2M Applications Using Diverse Communication Models

A 3GPP2 network operator may deploy the hybrid model with a combination of no internal and external M2M Server (as in the Direct Model) and internal and/or external M2M Server (as in the Indirect Model). As shown in figure C.2-2, a UE (an M2M Node such as ASN/MN) may be in communication with multiple M2M Servers which can be made up of a combination of 3GPP2 Underlying Network operator controlled and M2M Service Provider controlled M2M Servers. In that scenario, the M2M Service Provider controlled M2M Server, and the 3GPP2 Underlying Network operator controlled M2M Server may offer different capabilities to the M2M Applications.

Though not illustrated, it is also possible that in the Indirect Service Model with 3GPP2 network operator controlled M2M Server; the M2M Application may be inside the 3GPP2 network operator domain and under 3GPP network operator control.

C.3 3GPP2 Architectural Reference Model for M2M

Figure C.3-1 shows the architecture for a UE used for M2M connecting to the 3GPP2 Underlying Network. The architecture supports various architectural models described in clause C.2.

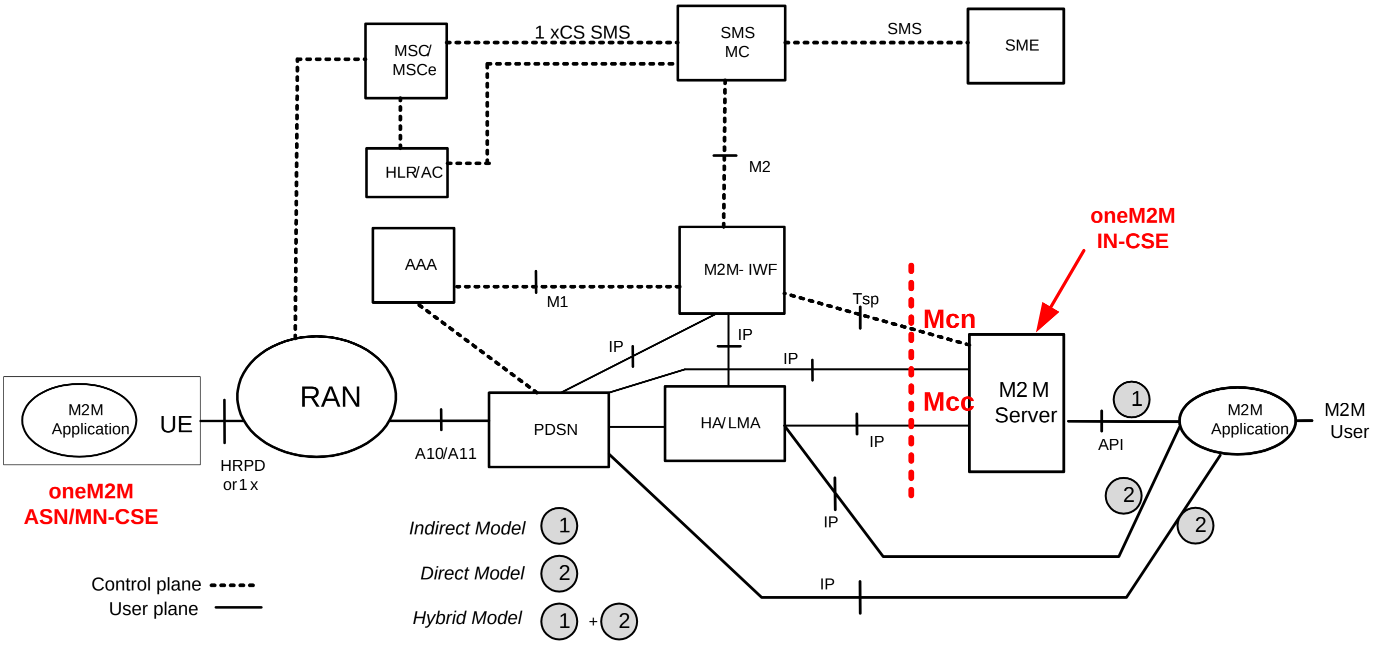


Figure C.3-1: Enhanced 3GPP2 Network Architecture for Supporting M2M

The M2M Server (IN-CSE) is the entity which connects to the 3GPP2 Underlying Network for providing communication with the UEs used for M2M. The M2M Server offers capabilities for use by one or multiple M2M Applications (AEs) hosted by the UE (ASN/MN). The corresponding M2M Applications in the external network (Infrastructure Domain) are hosted by one or multiple M2M Application platform(s).

The M2M Server interfaces with the 3GPP2 Underlying Network entities located in the home domain of the UE used for M2M via the Tsp and IP interfaces. M2M Server encompasses the IN-CSE entity specified by oneM2M. M2M Server interfaces with the M2M-IWF via Tsp Interface for Control Plane communications. User plane interactions between the M2M Server and 3GPP2 Underlying Network entities such as the PDSN and/or HA/LMA is via native-IP. With this configuration, oneM2M reference points Mcn and Mcc map to 3GPP2 reference points Tsp and IP respectively.

C.4 Communication between oneM2M Service Layer and 3GPP2 Underlying Network

Communication between the M2M Server (IN-CSE) and the entities in the 3GPP2 Underlying Network make use of the User Plane and the Control Plane communication paths, as needed for different 3GPP2 M2M communication models. User Plane communication path uses IP transport between the M2M Server (IN-CSE) and the CSE in the UE used for M2M (ASN/MN-CSE). The User Plane maps to oneM2M Mcc reference point. Control Plane communication path is over Tsp interface and maps to oneM2M Mcn reference point.

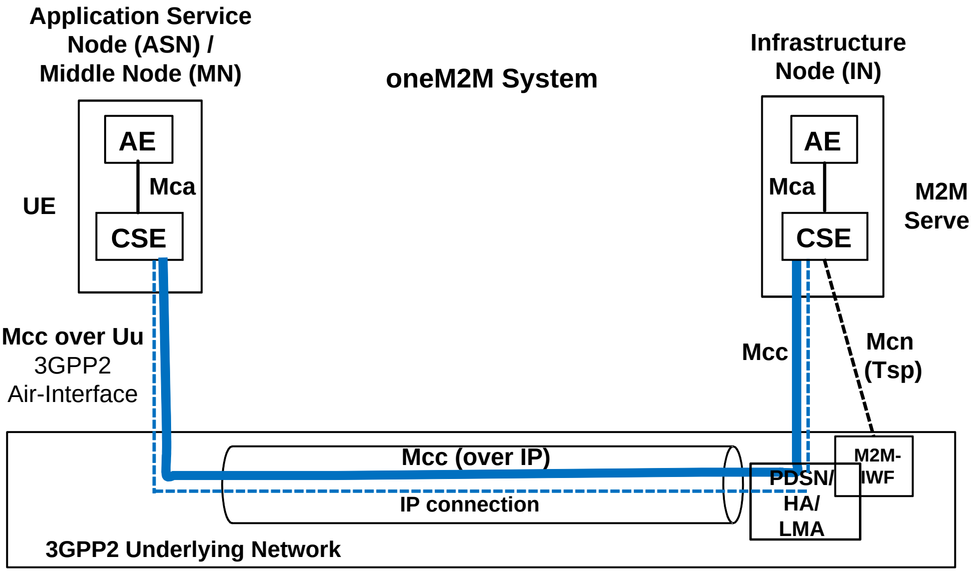


Figure C.4-1: User Plane and Control Plane Communication Paths

C.5 Information Flows

3GPP2 X.S0068 [i.20] specifies several system optimizations that can be used for M2M applications. Such optimizations include the following:

- Interaction of M2M Server with M2M-IWF for device triggering.
- Device trigger using SMS.
- Device trigger using broadcast SMS.
- Device trigger using IP transport.

C.5.1 Tsp Interface Call Flow

The following is the high level call flow illustrating device triggering using Tsp interface.

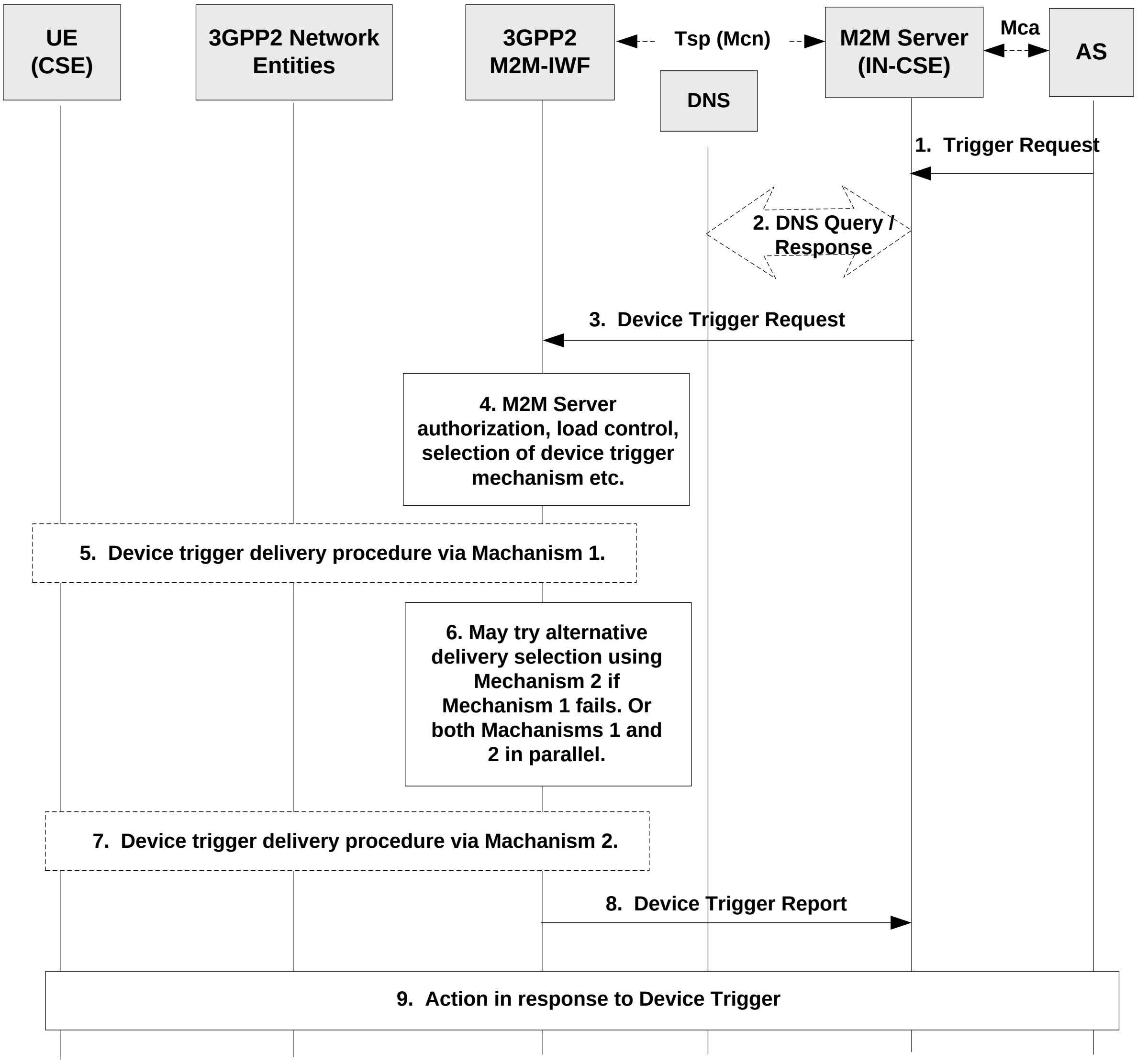

UE
(CSE)
3GPP2 Network Entities
3GPP2 M2M-IWF
Tsp (Mcn)
DNS
M2M Server (IN-CSE)
Mca
AS
1. Trigger Request
2. DNS Query / Response
3. Device Trigger Request
4. M2M Server authorization, load control, selection of device trigger mechanism etc.
5. Device trigger delivery procedure via Machanism 1.
6. May try alternative delivery selection using Mechanism 2 if Mechanism 1 fails. Or both Machanisms 1 and 2 in parallel.
7. Device trigger delivery procedure via Machanism 2.
8. Device Trigger Report
9. Action in response to Device Trigger

Figure C.5.1-1: Tsp Interface Call Flow

1) M2M Server (IN-CSE) receives a request from an M2M Application Server (AE in Infrastructure Domain) to deliver data to a UE used for M2M (ASN/MN-CSE) located in the 3GPP2 Underlying Network. Knowing the CSE-ID of the destination M2M Node, IN-CSE deduces its 3GPP2 External Identifier.

2) M2M Server (IN-CSE) may perform DNS query to obtain the IP address of the M2M-IWF for reaching the destination M2M Node.

3) M2M Server sends Device Trigger Request message to the M2M-IWF that includes destination M2M Node External ID and other information.

4) M2M-IWF checks that the M2M Server is authorized to send trigger requests and performs other tasks such as verifying that the M2M Server has not exceeded its quota or rate of trigger submission over Tsp. If such checks fail, the MTC-IWF sends a Device Trigger Confirm message with a cause value indicating the reason for the failure condition and the call flow stops at this step.

   Otherwise, the MTC-IWF continues to interact with HAAA/HLR for obtaining 3GPP2 Internal ID for the M2M Node and other information for reaching the M2M Node in the 3GPP2 Underlying Network. M2M-IWF also determines the device trigger mechanisms (e.g. Mechanism 1, Mechanism 2 etc.) supported by the M2M Node. The flow continues with Step 5.

5) M2M-IWF decides to deliver device trigger using e.g. Mechanism 1 and performs appropriate 3GPP2 Underlying Network specific procedures.

6) M2M-IWF may try alternative device trigger delivery mechanism (e.g. Mechanism 2) if Mechanism 1 fails. Or both Mechanism 1 and 2 can be performed in parallel.

7) M2M-IWF performs appropriate 3GPP2 Underlying Network specific procedures for delivering device trigger using Mechanism 2.

8) M2M-IWF sends Device Trigger Report to the M2M Server upon receiving the acknowledgment from the M2M Node that it has received M2M device trigger.

9) The M2M Node and the M2M Server/AS take actions in response to the device trigger as needed.

C.5.2 Point to Point Device Triggering

3GPP2 Underlying Network supports the following point-to-point device triggering mechanisms:

- SMS on common channel.
- SMS on 1xCS traffic channel.
- Device trigger using IP interface.

Device trigger using IP interface assumes that PPP sessions has been established and maintained between the M2M Node and the PDSN. An IP address has been assigned to the M2M Node by the IP anchor (PDSN/HA/LMA) and is maintained by the M2M Node and by other entities (e.g. HAAA) in the 3GPP2 Underlying Network. Upon receiving device trigger from the M2M Server, the M2M-IWF obtains the IP address assigned to the M2M Node from the M2M-AAA/HAAA. After that, the M2M-IWF sends device trigger to the M2M Node through IP routing via IP interface to the HA/LMA for MIP and PMIP operation, or to the PDSN for Simple IP operation.

C.5.3 Broadcast Device Triggering

3GPP2 Underlying Network supports the following broadcast device triggering mechanisms:

- SMS broadcast.

Annex D (normative): *<mgmtObj>* Resource Instances Description

D.1 oneM2M Management Functions

This clause describes the management functions supported by oneM2M. These functions are fulfilled by defining instances of *<mgmtObj>* resources. The instances are resources of *<mgmtObj>* resource type with specific designing to support different management capabilities through operations defined by oneM2M. The instances are service layer information models for the purpose of management. The instances can then be used within the M2M service layer or they can be further mapped to existing management technologies such as OMA DM [i.5], OMA LWM2M [i.6] and BBF TR-069 [i.4] to enable the management of devices with OMA or BBF compliant management clients.

NOTE: The resource instances defined in this clause are all of type *<mgmtObj>*. The instances are distinguished by attribute *mgmtDefinition*. The names of the instances are not fixed.

D.2 Resource *firmware*

The *[firmware]* resource is used to share information regarding the firmware on the device. The resource type of *[battery]* resource is *<mgmtObj>*resource.

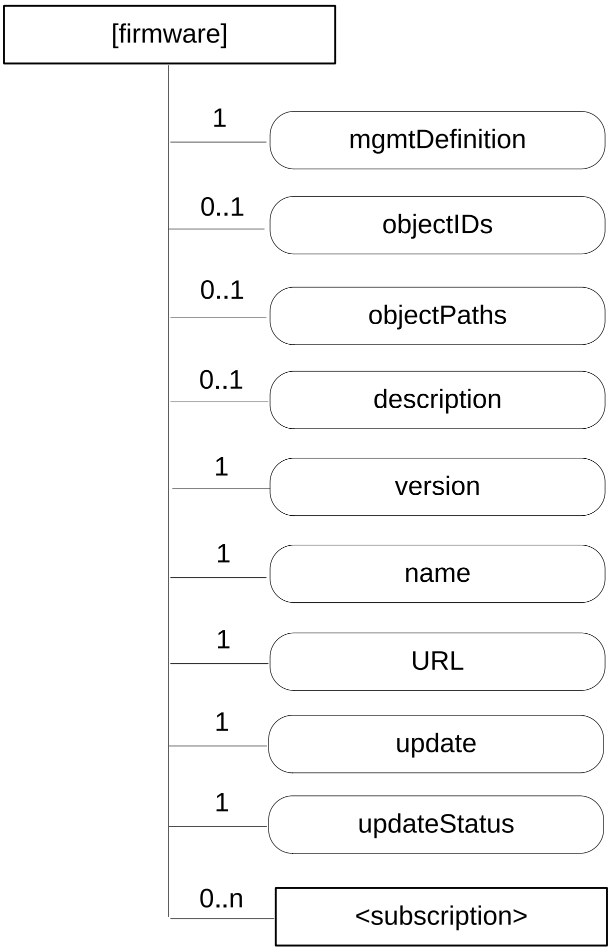


Figure D.2-1: Structure of *[firmware]* resource

The *[firmware]* resource shall contain the child resources specified in table D.2-1.

Table D.2-1: Child resources of *[firmware]* resource

Child Resources of *[firmware]*	Child Resource Type	Multiplicity	Description
[variable]	*<subscription>*	0..n	See clause 9.6.8 where the type of this resource is described.

The *[firmware]* resource shall contain the attributes specified in table D.2-2.

Table D.2-2: Attributes of *[firmware]* resource

Attributes of *[firmware]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	WO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described.
mgmtDefinition	1	WO	See clause 9.6.15. Has fixed value *"firmware"* to indicate the resource is for firmware management.
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
version	1	RW	The version of the firmware.
name	1	RW	The name of the firmware to be used on the device.
URL	1	RW	The URL from which the firmware image can be downloaded.
update	1	RW	The action that downloads and installs a new firmware in a single operation.
updateStatus	1	RO	Indicates the status of the update.

D.3 Resource *software*

The *[software]* resource is used to share information regarding the software on the device. The resource type of *[software]* resource is *<mgmtObj>*resource.

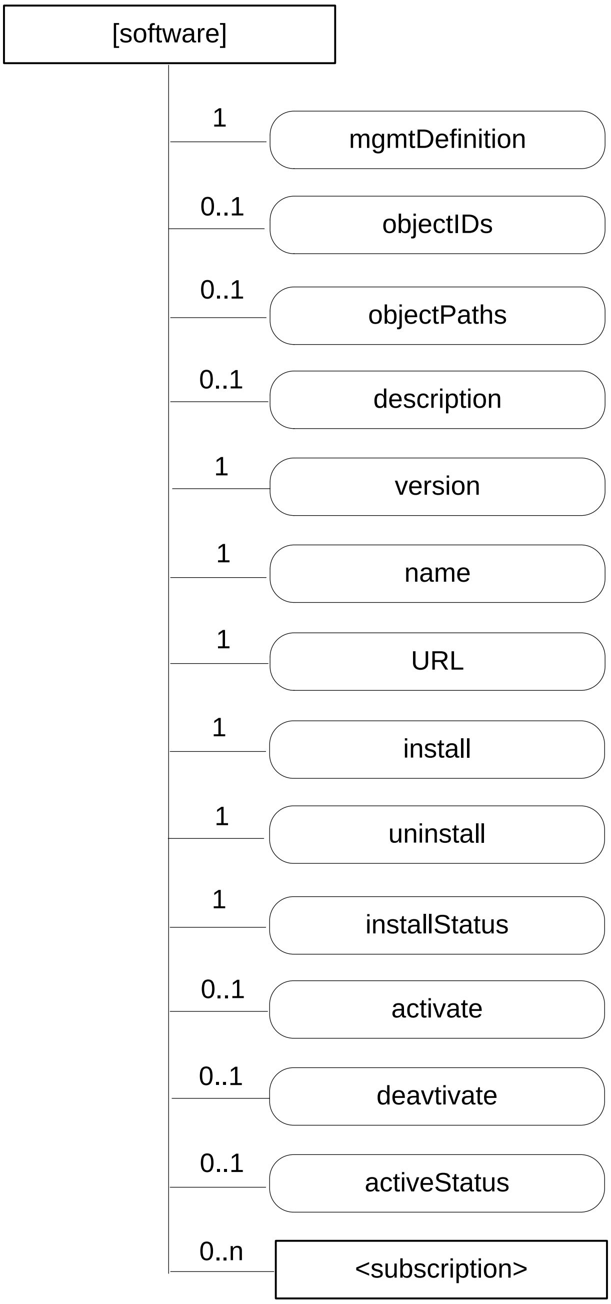


Figure D.3-1: Structure of *[software]* resource

The *[software]* resource shall contain the child resource specified in table D.3-1.

Table D.3-1: Child resources of *[software]* resource

Child Resources of *[software]*	Child Resource Type	Multiplicity	Description
[variable]	*<subscription>*	0..n	See clause 9.6.8 where the type of this resource is described.

The *[software]* resource shall contain the attributes specified in table D.3-2.

Table D.3-2: Attributes of *[software]* resource

Attributes of *[software]*	**Multiplicity**	**RW/ RO/ WO**	**Description**
resourceType	1	WO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described.
mgmtDefinition	1	WO	See clause 9.6.15. Has fixed value *"software"* to indicate the resource is for software management.
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
version	1	RW	The version of the software.
name	1	RW	The name of the software to be used on the device.
URL	1	RW	The URL from which the software package can be downloaded.
install	1	RW	The action that downloads and installs new software in a single operation.
uninstall	1	RW	The action that un-installs the software.
installStatus	1	RO	Indicates the status of the install.
activate	0..1	RW	The action that activates software previously installed.
deactivate	0..1	RW	The action that deactivates software.
activeStatus	0..1	RW	The status of active or deactivate action.

The state machine for managing the software in oneM2M is shown in figure D.3-2.

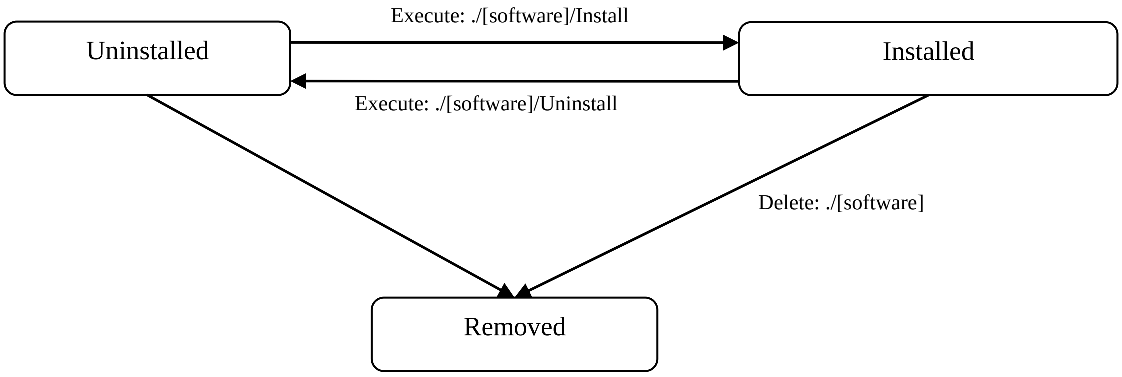

Execute: ./[software]/Install
Uninstalled
Installed
Execute: ./[software]/Uninstall
Delete: ./[software]
Removed

Figure D.3-2: State machine for *[software]* management

The following (figure D.3-3) is the state machine after install starts from the deactivated state.

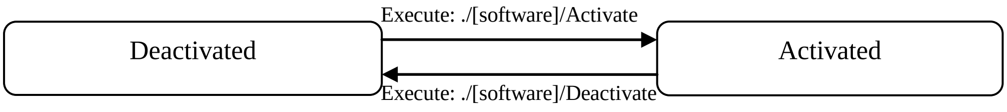

Execute: ./[software]/Activate
Deactivated
Activated
Execute: ./[software]/Deactivate

Figure D.3-3: State machine for *[software]* management after install

D.4 Resource *memory*

The *[memory]* resource is used to share information regarding the memory on the device. The resource type of *[memory]* resource is *<mgmtObj>*resource.

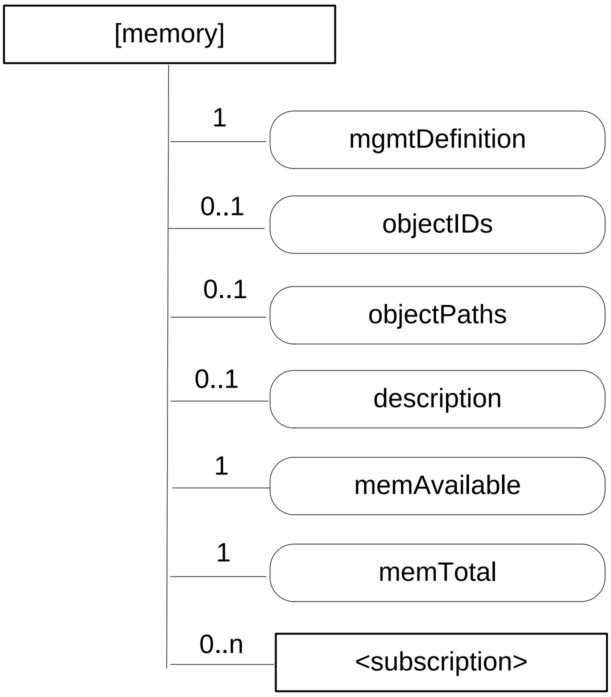


Figure D.4-1: Structure of *[memory]* resource

The *[memory]* resource shall contain the child resources specified in table D.4-1.

Table D.4-1: Child resources of *[memory]* resource

Child Resources of *[memory]*	Child Resource Type	Multiplicity	Description
[variable]	*<subscription>*	0..n	See clause 9.6.8 where the type of this resource is described.

The *[memory]* resource shall contain the attributes specified in table D.4-2.

Table D.4-2: Attributes of *[memory]* resource

Attributes of *[memory]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	WO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described.
mgmtDefinition	1	WO	See clause 9.6.15. Has fixed value *"memory"* to indicate the resource is for memory management.
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
memAvailable	1	RW	The current available amount of memory.
memTotal	1	RW	The total amount of memory.

D.5 Resource *areaNwkInfo*

The resource type of *[areaNwkInfo]* resource is *<mgmtObj>*resource.

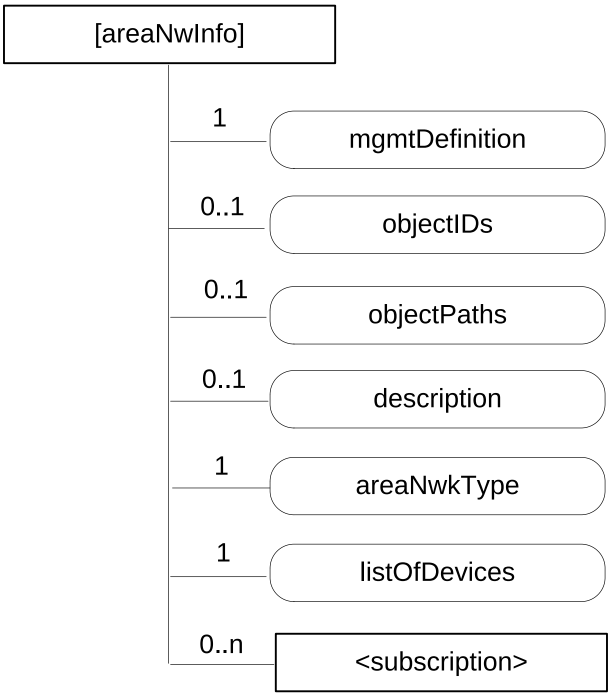


Figure D.5-1: Structure of *[areaNwkInfo]* resource

The *[areaNwkInfo]* resource shall contain the child resource specified in table D.5-1.

Table D.5-1: Child resources of *[areaNwInfo]* resource

Child Resources of *[areaNwkInfo]*	Child Resource Type	Multiplicity	Description
[variable]	*<subscription>*	0..n	See clause 9.6.8 where the type of this resource is described.

The *[areaNwkInfo]* resource shall contain the attributes specified in table D.5-2.

Table D.5-2: Attributes of *[areaNwkInfo]* resource

Attributes of *[areaNwkInfo]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	WO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described.
mgmtDefinition	1	WO	See clause 9.6.15. Has fixed value *"areaNwkInfo"* to indicate the resource is for area network information.
objectIDa	0..1	WO	See clause 9.6.15.
objectPatha	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
areaNwkType	1	RW	The *areaNwkType* is an implementation-chosen string that indicates the type of M2M Area Network.
listOfDevices	1	RW	Indicates the list of devices in the M2M Area Network. The attribute contains references to *[areaNwkDeviceInfo]* resource. From *listOfDevices*, the topology of the area network can be discovered and retrieved.

D.6 Resource *areaNwkDeviceInfo*

The resource type of *[areaNwkDeviceInfo]* resource is *<mgmtObj>*resource.

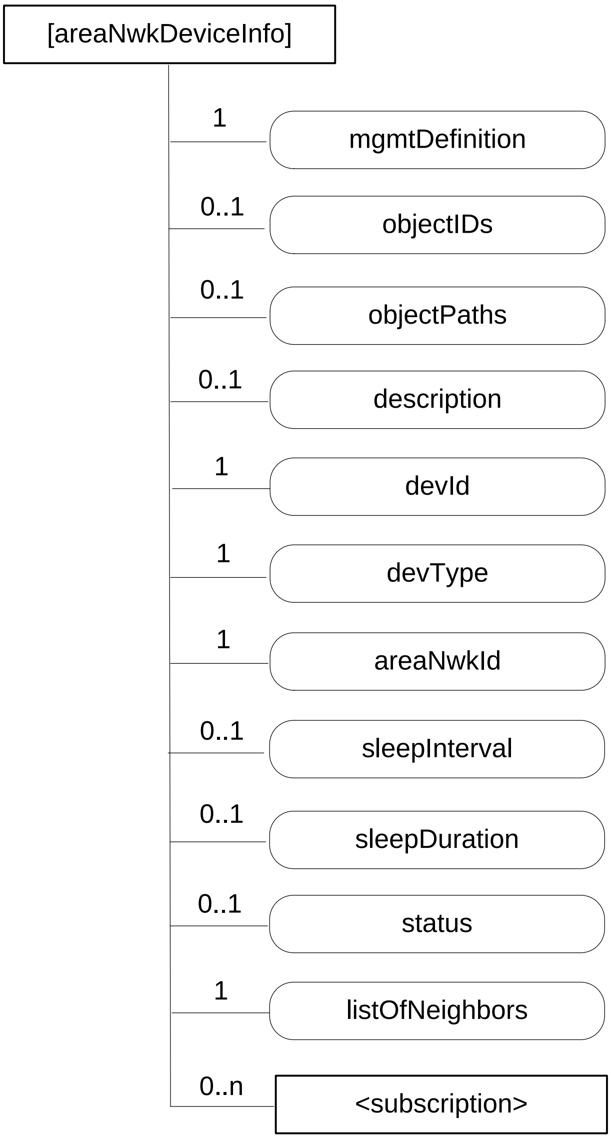


Figure D.6-1: Structure of *[areaNwkDeviceInfo]* resource

The *[areaNwkDeviceInfo]* resource shall contain the child resources specified in table D.6-1.

Table D.6-1: Child resources of *[areaNwkDeviceInfo]* resource

Child Resources of *[areaNwkDeviceInfo]*	Child Resource Type	Multiplicity	Description
[variable]	*<subscription>*	0..n	See clause 9.6.8 where the type of this resource is described.

The *[areaNwkDeviceInfo]* resource shall contain the attributes specified in table D.6-2.

Table D.6-2: Attributes of *[areaNwkDeviceInfo]* resource

Attributes of *[areaNwkDeviceInfo]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	WO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described.
mgmtDefinition	1	WO	See clause 9.6.15. Has fixed value *"areaNwkDeviceInfo"* to indicate the resource is for area network device information.
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
devId	1	RW	Indicates the id of the device. It could be the id of the hardware or *nodeId*.
devType	1	RW	Indicates the type of the device. The attribute also indicates the functions or services that are provided by the device. Examples include temperature sensor, actuator, Zigbee coordinator or Zigbee router.
areaNwkId	1	RW	The reference to an *areaNwkInfo* resource which this device associates with.
sleepInterval	0..1	RW	The interval between two sleeps.
sleepDuration	0..1	RW	The time duration of each sleep.
status	0..1	RW	The status of the device (sleeping or waked up).
listOfNeighbors	1	RW	Indicates the neighbour devices of the same area network. When modified, the connection relationship of the devices shall be modified accordingly.

D.7 Resource Type *battery*

The *[battery]* resource is used to share information regarding the battery. The resource type of *[battery]* resource is *<mgmtObj>* resource.

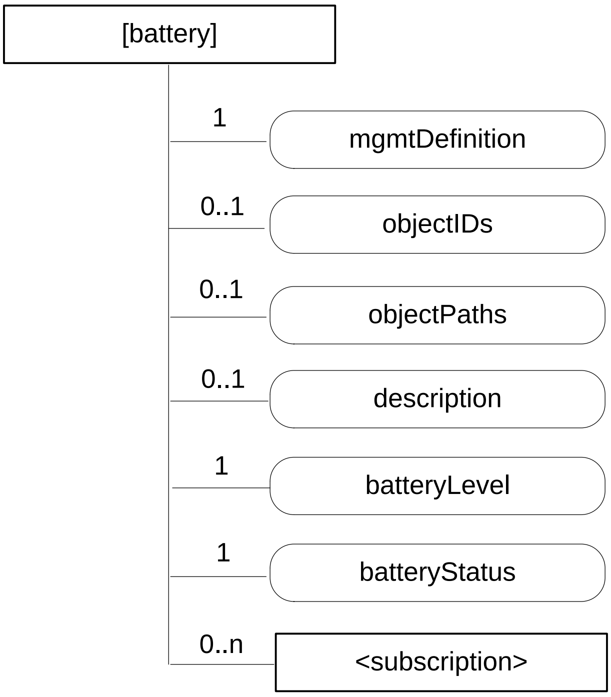


Figure D.7-1: Structure of *[battery]* resource

The *[battery]* resource shall contain the child resources specified in table D.7-1.

Table D.7-1: Child resources of *[battery]* resource

Child Resources of *[battery]*	Child Resource Type	Multiplicity	Description
[variable]	*<subscription>*	0..n	See clause 9.6.8 where the type of this resource is described.

The *[battery]* resource shall contain the attributes specified in table D.7-2.

Table D.7-2: Attributes of *[battery]* resource

Attributes of *[battery]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	WO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1		See clause 9.6.1 where this common attribute is described.
mgmtDefinition	1	WO	See clause 9.6.15. This attribute shall have the fixed value *"battery"*.
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
batteryLevel	1	RO	The current battery level.
batteryStatus	1	RO	Indicates the status of the battery.

D.8 Resource Type *deviceInfo*

The *[deviceInfo]* resource is used to share information regarding the device. The resource type of the *[deviceInfo]* resource is *<mgmtObj>* resource.

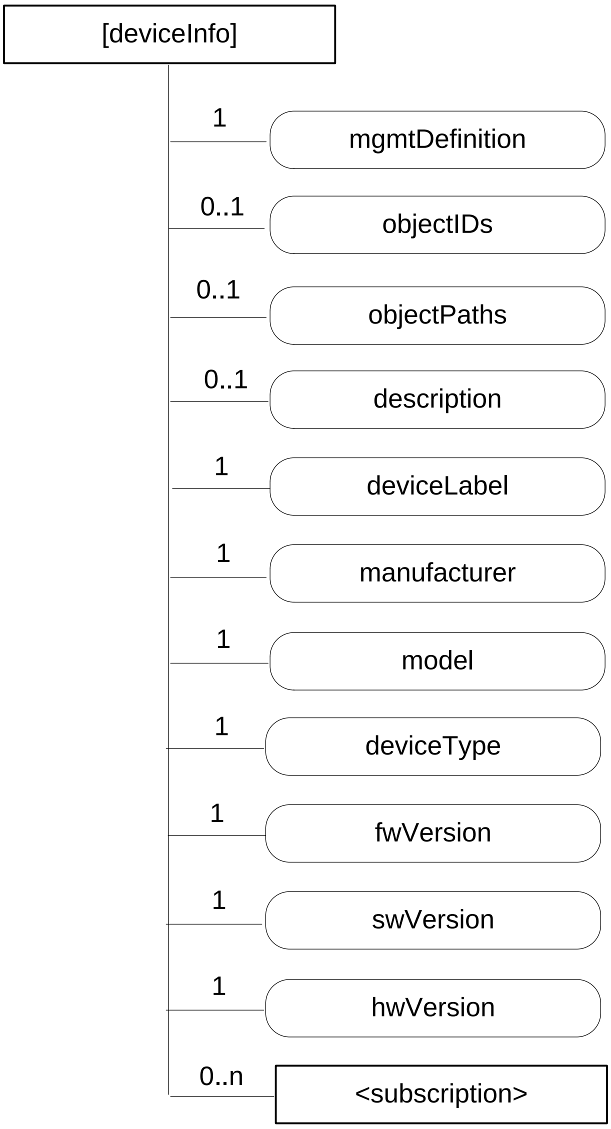


Figure D.8-1: Structure of *[deviceInfo]* resource

The *[deviceInfo]* resource shall contain the child resources specified in table D.8-1.

Table D.8-1: Child resources of *[deviceInfo]* resource

Child Resources of *[deviceInfo]*	Child Resource Type	Multiplicity	Description
[variable]	*<subscription>*	0..n	See clause 9.6.8 where the type of this resource is described.

The *[deviceInfo]* resource shall contain the attributes specified in table D.8-2.

Table D.8-2: Attributes of *[deviceInfo]* resource

Attributes of *[deviceInfo]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	WO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described.
mgmtDefinition	1	WO	See clause 9.6.15. This attribute shall have the fixed value *"deviceInfo"*.
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
deviceLabel	1	RO	Unique device label assigned by the manufacturer. The uniqueness may be global or only valid within a certain domain (e.g. vendor-wise or for a certain *deviceType*).
manufacturer	1	RO	The name/identifier of the device manufacturer.
model	1	RO	The name/identifier of the device mode assigned by the manufacturer.
deviceType	1	RO	The type (e.g. cell phone, photo frame, smart meter) or product class (e.g. X-series) of the device.
fwVersion	1	RO	The firmware version of the device. NOTE: If the device only supports one kind of Software this is identical to *swVersion*.
swVersion	1	RO	The software version of the device.
hwVersion	1	RO	The hardware version of the device.

D.9 Resource Type *deviceCapability*

The *[deviceCapability]* resource represents each device's capability. The resource type of *[deviceCapability]* resource is *<mgmtObj>* resource.

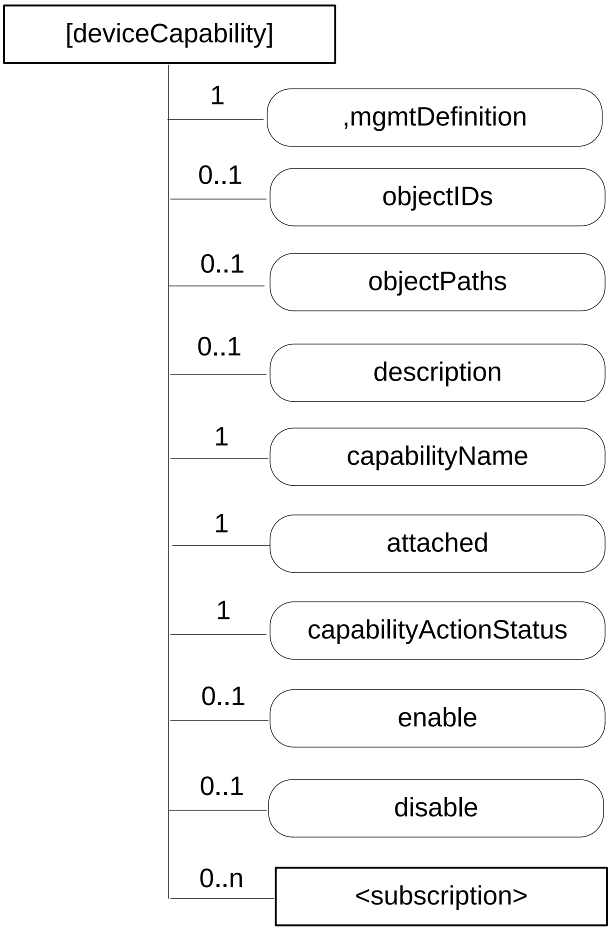


Figure D.9-1: Structure of *[deviceCapability]* resource

The *[deviceCapability]* resource shall contain the child resources specified in table D.9-1.

Table D.9-1: Child resources of *[deviceCapability]* resource

Child Resources of *[deviceCapability]*	Child Resource Type	Multiplicity	Description
[variable]	*<subscription>*	0..n	See clause 9.6.8 where the type of this resource is described.

The *[deviceCapability]* resource shall contain the attributes specified in table D.9-2.

Table D.9-2: Attributes of *[deviceCapability]* resource

Attributes of *[deviceCapability]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	WO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described.
mgmtDefinition	1	WO	See clause 9.6.15. This attribute shall have the fixed value *"deviceCapability"*.
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
capabilityName	1	WO	The name of the capability.
attached	1	RO	Indicates whether the capability is attached to the device or not.
capabilityActionStatus	1	RO	Indicates the status of the Action (including a progress indicator, a final state and a reminder of the requested action).
enable	0..1	WO	The action that allows enabling the device capability.
disable	0..1	WO	The action that allows disabling the device capability.

D.10 Resource Type *reboot*

The *[reboot]* resource is used to reboot a device. The resource type of *[reboot]* resource is *<mgmtObj>* resource.

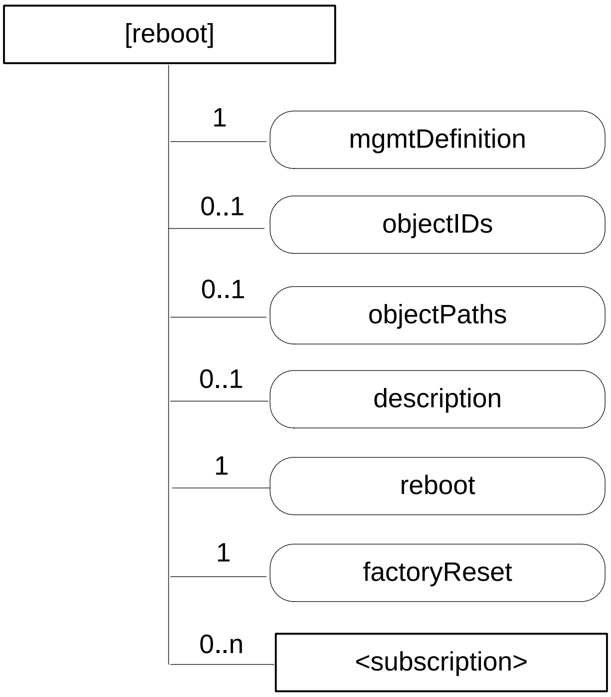


Figure D.10-1: Structure of *[reboot]* resource

The *[reboot]* resource shall contain the child resources specified in table D.10-1.

Table D.10-1: Child resources of *[reboot]* resource

Child Resources of *[reboot]*	Child Resource Type	Multiplicity	Description
[variable]	*<subscription>*	0..n	See clause 9.6.8 where the type of this resource is described.

The *[reboot]* resource shall contain the attributes specified in table D.10-2.

Table D.10-2: Attributes of *[reboot]* resource

Attributs of *[reboot]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	WO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described.
mgmtDefinition	1	WO	See clause 9.6.15. This attribute shall have the fixed value "reboot".
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
reboot	1	RW	The action that allows rebooting the device.
factoryReset	1	RW	The action that allows making the device returning to the factory settings.

D.11 Resource Type *eventLog*

The *[eventLog]* resource is used to record the event log for a device. The resource type of *[eventLog]* resource is *<mgmtObj>* resource.

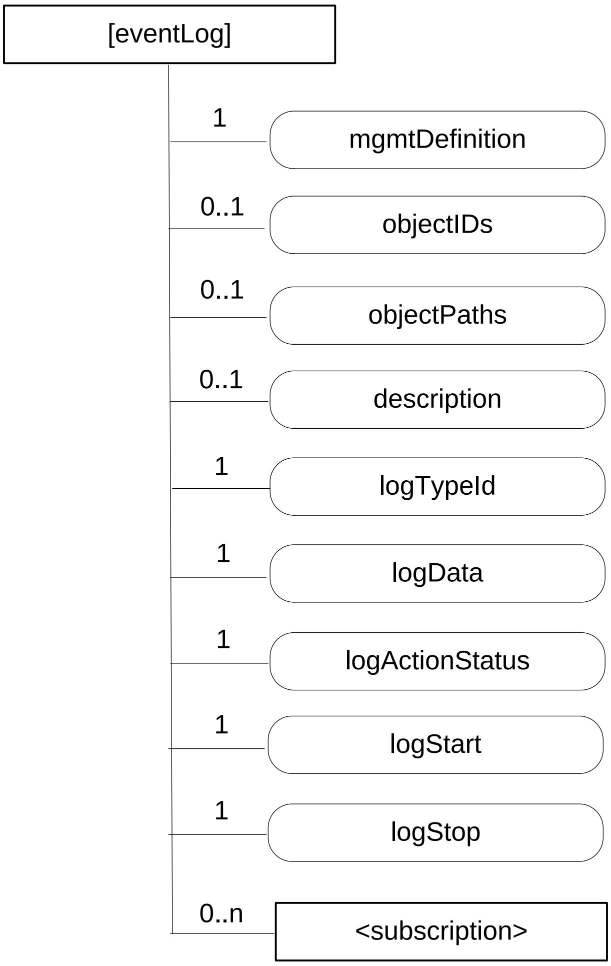


Figure D.11-1: Structure of *[eventLog]* resource

The *[eventLog]* resource shall contain the child resources specified in table D.11-1.

Table D.11-1: Child resources of *[eventLog]* resource

Child Resources of *[eventLog]*	Child Resource Type	Multiplicity	Description
[variable]	*<subscription>*	0..n	See clause 9.6.8 where the type of this resource is described.

The *[eventLog]* resource shall contain the attributes specified in table D.11-2.

Table D.11-2: Attributes of *[eventLog]* resource

Attributes of *[eventLog]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	WO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described.
mgmtDefinition	1	WO	See clause 9.6.15. This attribute shall have the fixed value *"eventLog"*.
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
logTypeId	1	RW	Identifies the types of log to be recorded. E.g. security log, system log.
logData	1	R	Diagnostic data logged upon event of interests defined by this diagnostic function.
logActionStatus	1	RO	Indicates the status of the Action. E.g. Started, Stopped.
logStart	1	RW	The action that allows starting the log corresponding to the mentioned *logTypeId*.
logStop	1	RW	The action that allows stopping the log corresponding to the mentioned *logTypeId*.

D.12 Resource *cmdhPolicy*

A *[cmdhPolicy]* resource is defined as a specialization of the *<mgmtObj>* resource type as specified in clause 9.6.15. It includes a number of child resources which are referenced by means of *mgmtLink* attributes. Each of these linked child resources represents itself a specialization of the *<mgmtObj>* resource type. These child resources and their child resources are defined in clauses D.12.1 to D.12.8.

The *[cmdhPolicy]* resource represents a set of rules associated with a specific CSE that govern the behaviour of that CSE regarding rejecting, buffering and sending request or response messages via the Mcc reference point. The rules contained in a *[cmdhPolicy]* resource are sub-divided into rules represented by different child resources with different purposes as follows:

- **Defaults:** Defines which CMDH related parameters will be used by default when a request or response message issued by a registrar of the associated CSE or the associated CSE itself contains the **ec** parameter but not all other CMDH related parameters and which default **ec** parameter shall be used when none is given in the request or response.

- **Limits:** Defines the allowed limits for CMDH related parameters in request or response messages with a given **ec** value.

- **Network usage:** Defines the conditions when usage of specific Underlying Networks is allowed for request or response messages with a given **ec** value.

- **Buffering:** Defines limits of supported buffer size to be used for storing pending messages with a given **ec** value and their priorities when deletion cannot be avoided.

The relationships of *[cmdhPolicy]* resources with other resources and the position within the overall resource structure are depicted in Figure D.12-1. One or several *[cmdhPolicy]* resources can be assigned as child resources under a parent of *<node>* resource type. The *<node>* resource carrying CMDH policies is linked by means of a *nodeLink* attribute from either the local *<CSEBase>* resource or an instance of a *<remoteCSE>* resource type. This *nodeLink* attribute as well as the reverse *hostedCSEID* attribute in the *<node>* resource define to which CSE the set of CMDH policies apply whenever this CSE receives requests or responses that need to be forwarded over Mcc reference point. Since only one particular set of CMDH rules can be active for a given CSE at any given point in time, an *[activeCMDHPolicy]* child resource under the parent <*node*> resource that represents the node which hosts the respective CSE is used to point to the active *[cmdhPolicy]* resource that shall be effective for that particular CSE.

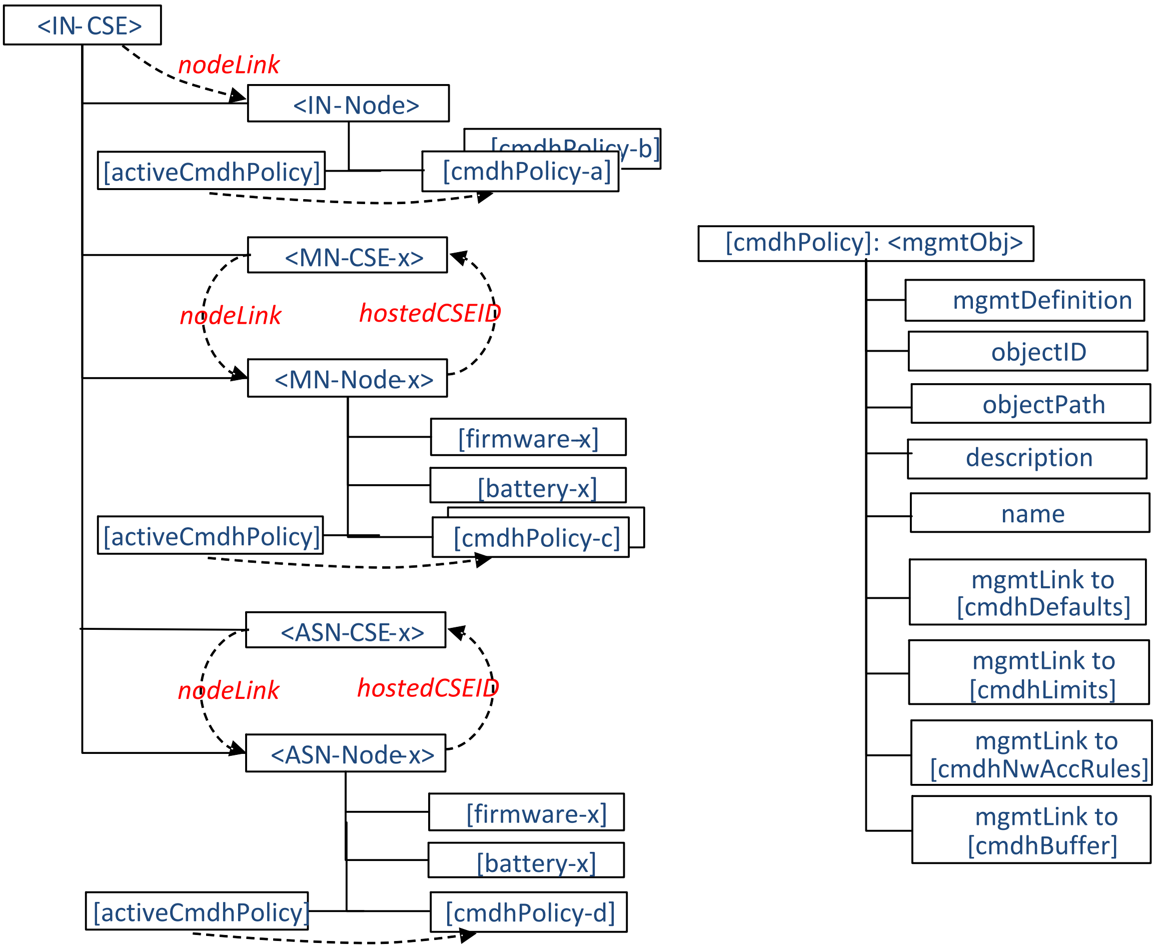


Figure D.12-1: Relationships between *[cmdhPolicy]* resource and other resources

When employing external management technology, the *[cmdhPolicy]* resources are assigned under instances of the <*node*> resources that represent the remotely managed field nodes in the IN-CSE performing device management for these nodes. In this scenario, the *[cmdhPolicy]* resources are transferred to the field node by means of the external device management technology applicable for that specific node.

When a field domain node is managed via the Mcc reference point, the *[cmdhPolicy]* resources are provisioned directly to instances of the <*node*> resources in the field domain CSE from an IN-CSE responsible for the device/entity management.

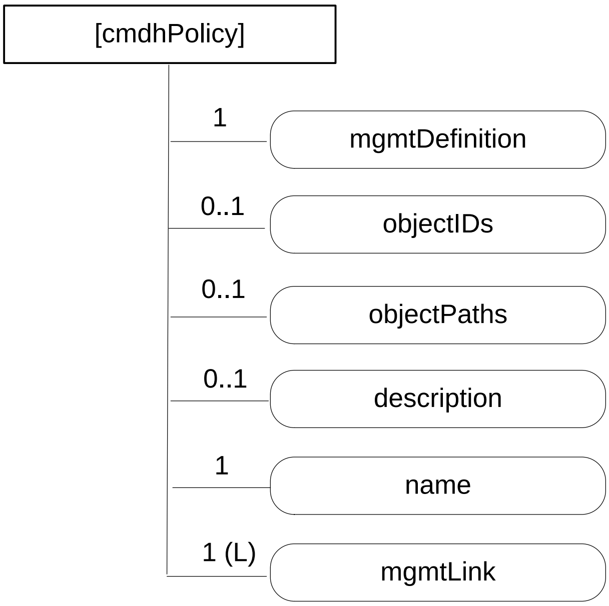


Figure D-12-2: Structure of *[cmdhPolicy]* resource

The *[cmdhPolicy]* resource shall contain attributes specified in table D.12-1.

Table D.12-1: Attributes of *[cmdhPolicy]* resource

Attributes of *[cmdhPolicy]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described.
mgmtDefinition	1	WO	See clause 9.6.15. Has fixed value *"cmdhPolicy"* to indicate the resource is for CMDH policy management.
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
name	1	RW	A name under which the CMDH policy will be referred.
mgmtLink	1 (L)	RW	A list containing at least 4 links. • 1 link to *[cmdhDefaults]* resource; • At least 1 or more link(s) to *[cmdhLimits]* resource(s); • At least 1 or more link(s) to *[cmdhNetworkAccessRules]* resource(s); • At least 1 or more link(s) to *[cmdhBuffer]* resource(s).

D.12.1 Resource *activeCmdhPolicy*

A managed node can have one or more sets of *[cmdhPolicy]* resources assigned as children.

The *[activeCmdhPolicy]* resource is used to provide a link to the currently active set of CMDH policies. This identifies which set of CMDH policies is currently actively in use in the corresponding CSE. It allows the device management technology to activate a policy set independently of the download of a new set of CMDH policies in order to avoid potential race conditions. The *[activeCmdhPolicy]* and *[cmdhPolicy]* resources are children of the same *<node>* resource to which these policies apply.

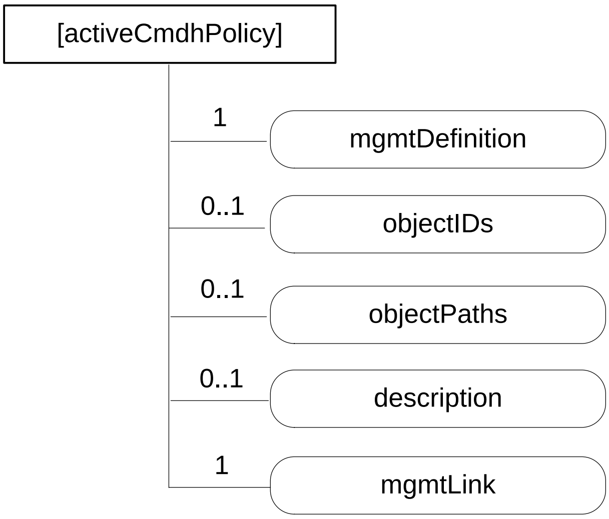


Figure D.12.1-1: Structure of *[activeCmdhPolicy]* resource

The *[activeCmdhPolicy]* resource shall contain attributes specified in table D.12.1-1.

Table D.12.1-1: Attributes of *[activeCmdhPolicy]* resource

Attributes of *[activeCmdhPolicy]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described
mgmtDefinition	1	WO	See clause 9.6.15. Has fixed value *"activeCmdhPolicy".*
objectIDs	0..1	WO	See section 9.6.15.
objectPaths	0..1	WO	See section 9.6.15.
description	0..1	RW	See section 9.6.15.
mgmtLink	1	RW	link to active *[cmdhPolicy]* resource.

D.12.2 Resource *cmdhDefaults*

The *[cmdhDefaults]* resource is used to define default values that shall be used for CMDH-related parameters when requests issued by originators (registered AEs or functions inside the CSE itself) do not contain a value for the parameters **ec** (event category), **rqet** (request expiration timestamp), **rset** (result expiration timestamp), **oet** (operational execution time), **rp** (response persistence), and/or **da** (delivery aggregation).

Upon receiving a request, the CSE will first look if the **ec** (event category) parameter is set. If not, it will use the *[cmdhDefEcValue]* resources (see below) to determine a value that should be used for **ec**.

Then, if any of the parameters **rqet**, **rset**, **oet**, **rp** or **da** is not set, the CSE will use the *[cmdhEcDefParamValues]* resources (see below) to populate the missing parameters (and only the missing ones).

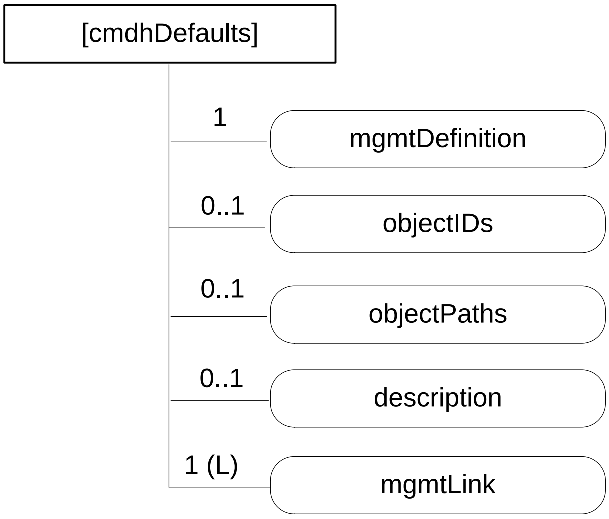


Figure D.12.2-1: Structure of *[cmdhDefaults]* resource

The *[cmdhDefaults]* resource shall contain attributes specified in table D.12-2-1.

Table D.12.2-1: Attributes of *[cmdhDefaults]* resource

Attributes of *[cmdhDefaults]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described
mgmtDefinition	1	WO	See clause 9.6.15. Has fixed value *"cmdhDefaults"*.
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
mgmtLink	1 (L)	RW	A list containing at least 2 links: • At least 1 or more link(s) to *[cmdhDefEcValue]* resource(s); • At least 1 or more link(s) to *[cmdhEcDefParamValues]* resource(s).

D.12.3 Resource *cmdhDefEcValue*

The *[cmdhDefEcValue]* resource is used to define a value for the **ec** (event category) parameter of an incoming request when it is not defined.

Upon receiving a request, the CSE will go through all the *[cmdhDefEcValue]* resources (in the order of their *order* attribute), check the *requestOrigin* and any present *requestContext* and *requestCharacteristics* attributes to see if they match (see description of matching), and if they all do, assign the value stored in the *defEcValue* attribute to the **ec** parameter.

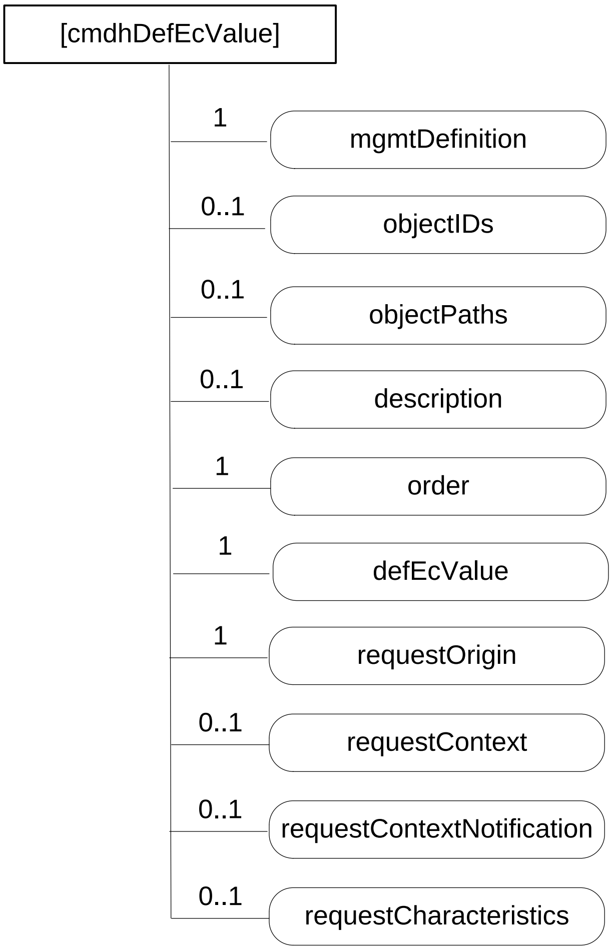


Figure D.12.3-1: Structure of *[cmdhDefEcValue]* resource

The *[cmdhDefEcValue]* resource shall contain attributes specified in table D.12.3-1.

Table D.12.3-1: Attributes of *[cmdhDefEcValue]* resource

Attributes of *[cmdhDefEcValue]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described
mgmtDefinition	1	WO	See clause 9.6.15. Has fixed value *"cmdhDefEcValue"*.
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
order	1	RW	The index indicating in which order the *[cmdhDefEcValue]* resource will be treated by the CSE to determine a value for the **ec** parameter
defEcValue	1	RW	The actual value to use for the **ec** parameter if the conditions expressed in the *requestOrigin*, *requestContext* and *requestCharacteristics* attributes all match. If none of these attributes are defined, then the *defEcValue* shall be applied.
requestOrigin	1	RW	The *requestOrigin* attribute is a list of zero or more local *AE-IDs*, *App-IDs*, or the strings 'localAE' or 'thisCSE'. When an *AE-ID* appears in the *requestOrigin* attribute, the default **ec** value defined inside the *defEcValue* attribute is applicable for the **ec** if the request was issued by that specific Application Entity. When an *App-ID* appears in the *requestOrigin* attribute, the default **ec** value defined inside the *defEcValue* attribute is applicable for the **ec** if the request was issued by the AE with that *App-ID* unless covered by another *[cmdhDefEcValue]* resource with a *requestOrigin* attribute containing its specific *AE-ID*. When the string 'localAE' appears in the *requestOrigin* attribute, the default **ec** value defined inside the *defEcValue* attribute is applicable for the **ec** for requests issued by all local AEs unless covered by another *[cmdhDefEcValue]* resource with a *requestOrigin* attribute containing the specific *AE-ID* or *App-ID* of the originator of the request. When the string 'thisCSE' appears in the *requestOrigin* attribute, the default **ec** value defined inside the *defEcValue* attribute is applicable for the **ec** for requests that are originating from within the registrar CSE. The hosting CSE shall contain at least one *[cmdhDefEcValue]* resource that contains 'localAE' in the *requestOrigin* attribute and has no *requestContext* and no *requestCharacteristics* attribute. The hosting CSE shall contain at least one *[cmdhDefEcValue]* resource that contains 'thisCSE' in the *requestOrigin* attribute and has no *contextCondtion* and no *requestCharacteristics* attribute.
requestContext	0..1	RW	The *requestContext* attribute represents the Dynamic Context condition under which the default **ec** value defined inside the *defEcValue* attribute is applicable for the **ec**. This may refer to conditions such as current battery status, or current network signal strength.

Attributes of *[cmdhDefEcValue]*	Multiplicity	RW/ RO/ WO	Description
requestContextNotification	0..1	RW	True or false. If set to true, then this CSE will establish a subscription to the dynamic context information defined in the *requestContext* attribute as well as a subscription to this *[cmdhDefEcValue]* resource for all AEs corresponding to the *AE-ID* or an *App-ID* appearing in the *requestOrigin* attribute. Both, changes in the context information and changes to the *[cmdhDefEcValue]* resource will be notified to the respective AEs. The subscription(s) is/are established when the *[cmdhDefEcValue]* is provisioned or updated.
requestCharacteristics	0..1	RW	The *requestCharacteristics* attribute represents conditions pertaining to the request itself, such as the requested response type (**rt** parameter) or other parameters of the request.

D.12.4 Resource *cmdhEcDefParamValues*

The *[cmdhEcDefParamValues]* resource is used to represent a specific set of default values for the CMDH related parameters **rqet** (request expiration timestamp), **rset** (result expiration timestamp), **oet** (operational execution time), **rp** (response persistence) and **da** (delivery aggregation) that are applicable for a given **ec** (event category) if these parameters are not specified in the request.

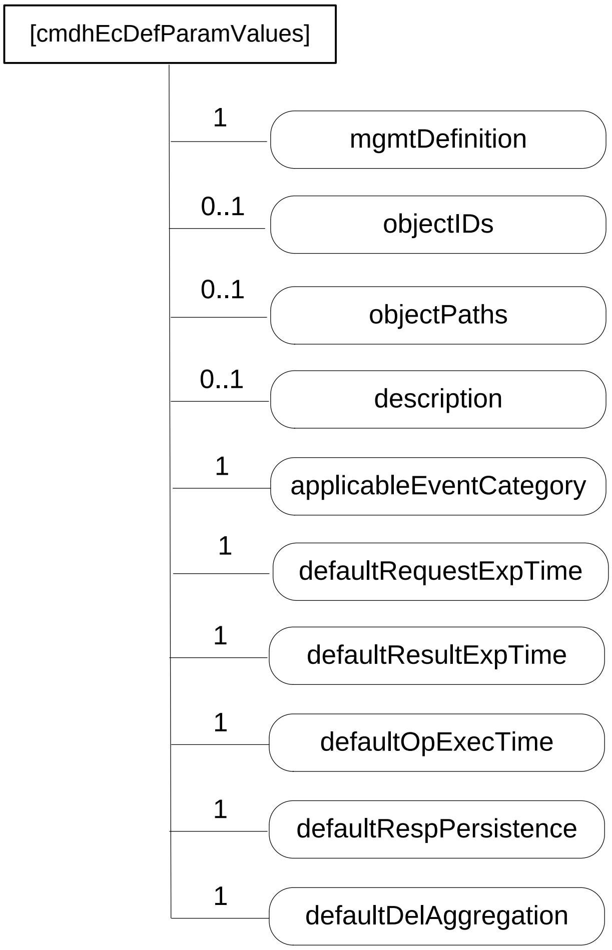


Figure D.12.4-1: Structure of *[cmdhEcDefParamValues]* resource

The *[cmdhEcDefParamValues]* resource shall contain attributes specified in table D.12.4-1.

Table D.12.4-1: Attributes of *[cmdhEcDefParamValues]* resource

Attributes of *[cmdhEcDefParamValues]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described
mgmtDefinition	1	WO	See clause 9.6.15. Has fixed value *"cmdhEcDefParamValues"*.
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
applicableEventCategory	1	RW	This attribute defines the event categories for which this set of default parameters defined in this *[cmdhEcDefParamValues]* resource are applicable. This attribute is a list of zero or more Event Category values (**ec** parameter of a request), or the string 'default'. When an Event Category value appears in the *applicableEventCategory* attribute, the set of default parameters defined in this *[cmdhEcDefParamValues]* resource are applicable for requests associated with that specific Event Category (**ec**) value. When the string 'default' appears in the *applicableEventCategory* attribute, the set of default parameters defined in this *[cmdhEcDefParamValues]* resource are applicable for all requests whose associated Event Category value (**ec**) is not listed in the *applicableEventCategory* attribute of any other provisioned *[cmdhEcDefParamValues]* resource on the hosting CSE. A specific Event Category value (**ec**) shall appear at most once in any of the *applicableEventCategory* attributes of any of the provisioned *[cmdhEcDefParamValues]* resources on the hosting CSE. The string 'default' shall appear exactly once in any of the *applicableEventCategory* attributes of any of the provisioned *[cmdhEcDefParamValues]* resources on the hosting CSE.
defaultRequestExpTime	1	RW	Default value for the request expiration time parameter (**rqet**) in a request when the **rqet** parameter of the request is not set.
defaultResultExpTime	1	RW	Default value for the result expiration time parameter (**rset**) in a request when the **rset** parameter of the request is not set.
defaultOpExecTime	1	RW	Default value for the operational execution time parameter (**oet**) in a request when the **oet** parameter of the request is not set.
defaultRespPersistence	1	RW	Default value for the response persistence parameter (**rp**) in a request when the **rp** parameter of the request is not set.
defaultDelAggregation	1	RW	Default value for the delivery aggregation parameter (**da**) in a request when the **da** parameter of the request is not set.

D.12.5 Resource *cmdhLimits*

The *[cmdhLimits]* resource is used to define limits for CMDH related parameter values used in requests issued by originators (registered AEs or functions inside the CSE itself). When an incoming request is processed that does not comply with the limits defined by the corresponding *[cmdhLimits]* resource, the request shall be rejected by the CSE.

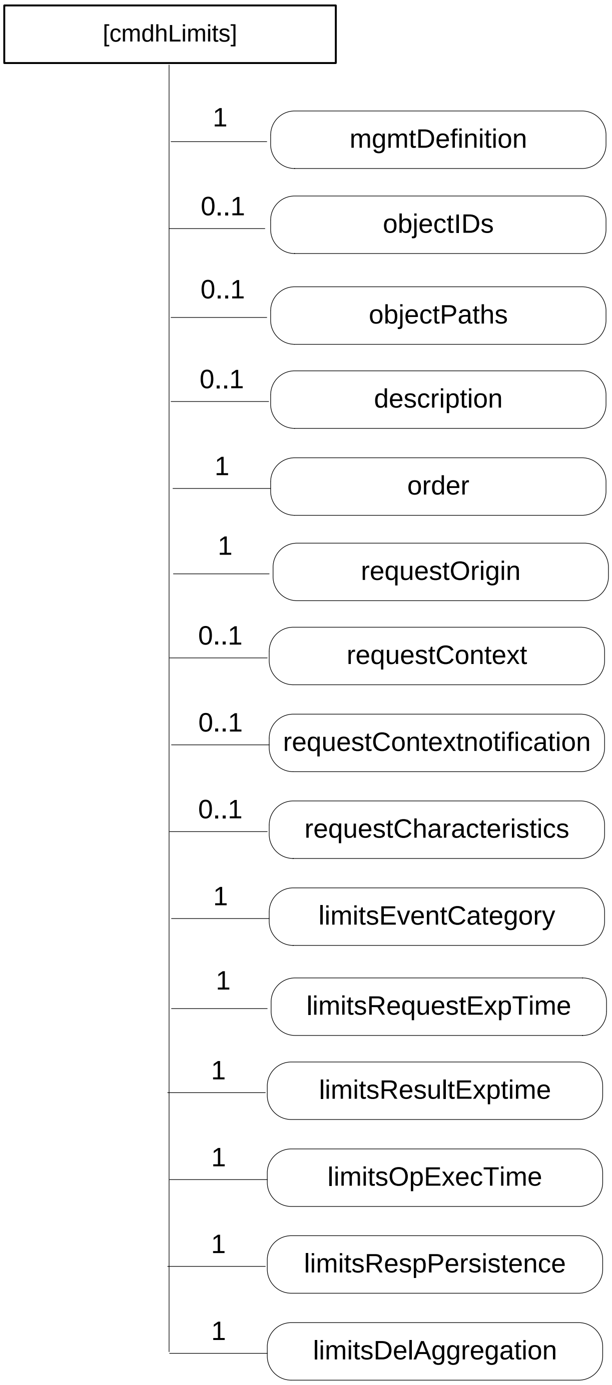


Figure D.12.5-1: Structure of *[cmdhLimits]* resource

The *[cmdhLimits]* resource shall contain attributes specified table D.12.5-1.

Table D.12.5-1: Attributes of *[cmdhLimits]* resource

Attributes of *[cmdhLimits]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described
mgmtDefinition	1	WO	See clause 9.6.15. Has fixed value *"cmdhLimits"*.
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
order	1	RW	The index indicating in which order the *[cmdhLimits]* resource will be treated by the CSE to determine a value for the limit parameters.
requestOrigin	1	RW	The *requestOrigin* attribute is a list of zero or more local *AE-IDs*, *App-IDs*, or the strings 'localAE' or 'thisCSE'. When an *AE-ID* appears in the *requestOrigin* attribute, the CMDH parameter limits defined inside *[cmdhLimits]* resources are applicable for requests issued by that specific Application Entity. When an *App-ID* appears in the *requestOrigin* attribute, the CMDH parameter limits defined inside *[cmdhLimits]* resources are applicable for requests issued by the AE with that *App-ID* unless already covered by another *[cmdhLimits]* resource with a *requestOrigin* attribute containing its specific *AE-ID*. When the string 'localAE' appears in the *requestOrigin* attribute, CMDH parameter limits defined inside *[cmdhLimits]* resources are applicable for all local AEs unless covered by another *[cmdhLimits]* resource with a *requestOrigin* attribute containing the specific *AE-ID* or *App-ID* of the originator of the request. When the string 'thisCSE' appears in the *requestOrigin* attribute, CMDH parameter limits defined inside *[cmdhLimits]* resources are applicable for all requests that are originating from within the hosting CSE. The hosting CSE shall contain at least one *[cmdhLimits]* resource that contains 'localAE' in the *requestOrigin* attribute and has no *contextCondition* and no *requestCharacteristics* attribute. The hosting CSE shall contain at least one *[cmdhLimits]* resource that contains 'thisCSE' in the *requestOrigin* attribute and has no *requestContext* and no *requestCharacteristics* attribute.
requestContext	0..1	RW	The *requestContext* attribute represents the Dynamic Context condition under which CMDH parameter limits defined inside the *[cmdhLimits]* resource is applicable. This may refer to conditions such as current battery status, or current network signal strength.

Attributes of *[cmdhLimits]*	**Multiplicity**	**RW/ RO/ WO**	**Description**
requestContextNotification	0..1	RW	True or false. If set to true, then this CSE will establish a subscription to the dynamic context information defined in the *requestContext* attribute as well as a subscription to this *[cmdhLimits]* resource for all AEs corresponding to the *AE-ID* or an *App-ID* appearing in the *requestOrigin* attribute. Both, changes in the context information and changes to the *[cmdhLimits]* resource will be notified to the respective AEs. The subscription(s) is/are established when the *[cmdhLimits]* is provisioned or updated
requestCharacteristics	0..1	RW	The *requestCharacteristics* attribute represents conditions pertaining to the request itself, such as the requested response type (**rt** attribute) or other attributes of the request.
limitsEventCategory	1	RW	Allowed values for the event category parameter (**ec**) in a request of any of the originators indicated in the *requestOrigin* attribute.
limitsRequestExpTime	1	RW	Range of allowed values for the request expiration time parameter (**rqet**) in a request of any of the originators indicated in the *requestOrigin* attribute.
limitsResultExpTime	1	RW	Range of allowed values for the result expiration time parameter (**rset**) in a request of any of the originators indicated in the *requestOrigin* attribute.
limitsOpExecTime	1	RW	Range of allowed values for the operational execution time parameter(**oet**) in a request of any of the originators indicated in the *requestOrigin* attribute.
limitsRespPersistence	1	RW	Range of allowed values for the response persistence parameter (**rp**) in a request of any of the originators indicated in the *requestOrigin* attribute.
limitsDelAggregation	1	RW	List of allowed values for the delivery aggregation parameter (**da**) in a request of any of the originators indicated in the *requestOrigin* attribute.

D.12.6 Resource *cmdhNetworkAccessRules*

The *[cmdhNetworkAccessRules]* resource is used to define the usage of Underlying Networks for forwarding information to other CSEs during processing of CMDH-related requests in a CSE. When an incoming request is processed by a CSE, it can only use Underlying Networks for forwarding any information to other CSEs in compliance with the rules defined by the corresponding *[cmdhNetworkAccessRules]* resource.

If a request cannot be successfully completed in compliance with the rules defined in the corresponding *[cmdhNetworkAccessRules]* resource, that request shall either be rejected in case it has not already been accepted by the CSE or it has to be purged. Error reporting on failed CMDH processing depends on error reporting parameters (TBD).

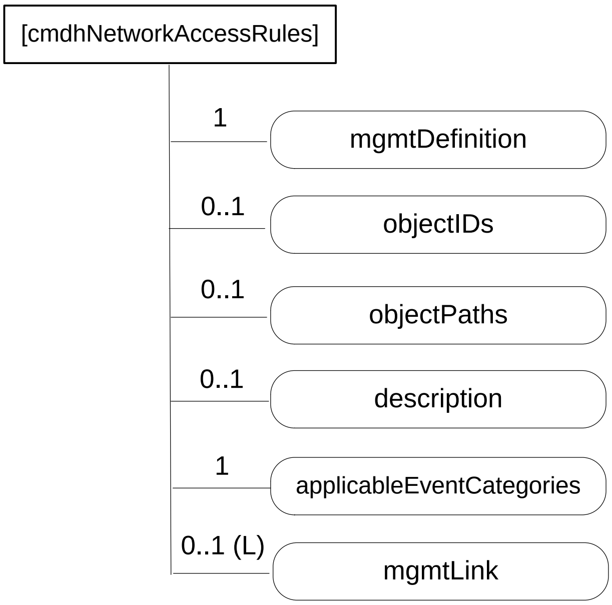


Figure D.12.6-1: Structure of *[cmdhNetworkAccessRules]* resource

If a *[cmdhNetworkAccessRules]* resource has no *mgmtLink* attribute to *[cmdhNwAccessRules]* resources (i.e. multiplicity of 0), requests that match with the *applicableEventCategori*e attribute (see description of attributes in table D.12.6-1) will not be allowed to use any Underlying Network for forwarding information, i.e. such requests need to be rejected.

The *[cmdhNetworkAccessRules]* resource shall contain attributes specified in table D.12.6-1.

Table D.12.6-1: Attributes of *[cmdhNetworkAccessRules]* resource

Attributes of *[cmdhNetworkAccessRules]*	**Multiplicity**	**RW/ RO/ WO**	**Description**
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described
mgmtDefinition	1	WO	See clause 9.6.15. Has fixed value "*cmdhNetworkAccessRules*".
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
applicableEventCategories	1	RW	This attribute defines for which requests the rules contained in *[cmdhNwAccessRule]* resources linked from this *[cmdhNetworkAccessRules]* resource shall be applied. This attribute is a list of zero or more Event Category values (**ec** parameter of a request), or the string 'default'. When an Event Category value appears in the *applicableEventCategories* attribute, the network usage rules defined inside *[cmdhNwAccessRule]* child resources are applicable for requests associated with that specific Event Category (**ec**) value. When the string 'default' appears in the *applicableEventCategories* attribute, the network usage rules defined inside *[cmdhNwAccessRule]* child resources are applicable for all requests whose associated Event Category value (**ec**) is not listed in the *applicableEventCategories* attribute of any other provisioned *[cmdhNetworkAccessRules]* resource on the hosting CSE. A specific Event Category value (**ec**) shall appear at most once in any of the *applicableEventCategories* attributes of any of the provisioned *[cmdhNetworkAccessRules]* resources on the hosting CSE. The string 'default' shall appear exactly once in any of the *applicableEventCategories* attributes of any of the provisioned *[cmdhNetworkAccessRules]* resources on the hosting CSE.
mgmtLink	0..1 (L)	RW	List of link(s) to *[cmdhNwAccessRule]* resource(s)

D.12.7 Resource *cmdhNwAccessRule*

The *[cmdhNwAccessRule]* resource is used define limits in usage of specific Underlying Networks for forwarding information to other CSEs during processing of CMDH-related requests.

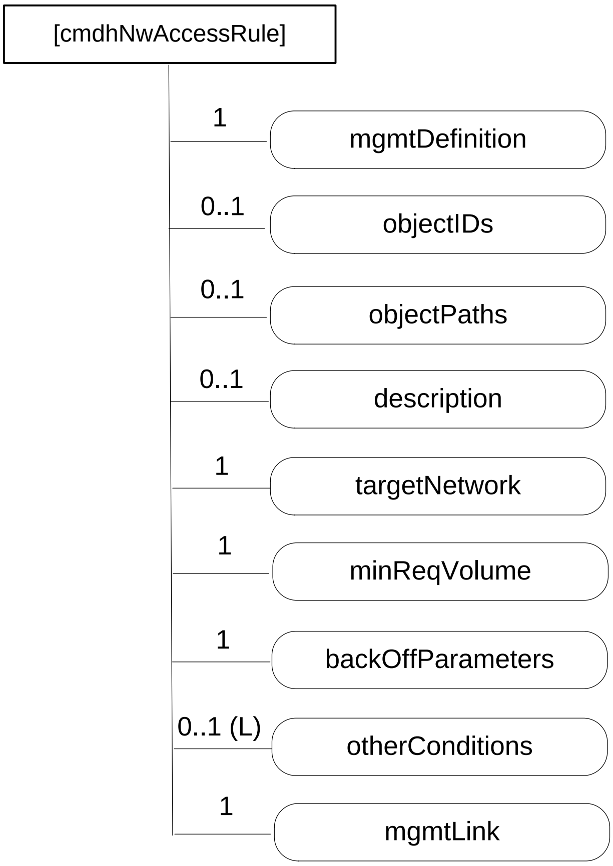


Figure D.12.7-1: Structure of *[cmdhNwAccessRule]* resource

The <*allowedSchedule*> child resource defines the periods of time during which it is allowed to use the Underlying Networks that match with the *targetNetwork* attribute of this *[cmdhNwAccessRule]* resource (see description of attributes in table D.12.7-1) when forwarding information to other CSEs due to requests that match the *applicableEventCategories* attribute of the parent *[cmdhNetworkAccessRules]* resource of this *[cmdhNwAccessRule]* resource.

The *[cmdhNwAccessRule]* resource shall contain attributes specified in table D.12.7-1.

Table D.12.7-1: Attributes of *[cmdhNwAccessRule]* resource

Attributes of *[cmdhNwAccessRule]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described
mgmtDefinition	1	WO	See clause 9.6.15. Has fixed value *"cmdhNwAccessRules"*.
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
targetNetwork	1	RW	The *targetNetwork* attribute defines for which Underlying Networks the usage limits contained in this *[cmdhNwAccessRule]* resource shall be applied. The *targetNetwork* attribute is a list of one or more strings identifying names of Underlying Networks or the string 'default'. NOTE: An agreed naming convention for Underlying Network names are TBD. When a name of an Underlying Network appears in the *targetNetwork* attribute, the usage limits contained in this *[cmdhNwAccessRule]* resource shall be applied for usage of that specific Underlying Network when processing requests matching with the parent *[cmdhNetworkAccessRules]* resource's *applicableEventCategories* attribute. When the string 'default' appears in the *targetNetwork* attribute, the usage limits contained in this *[cmdhNwAccessRule]* resource shall be applied for usage of all Underlying Networks that are not listed with their specific name in the *targetNetwork* attribute of any other *[cmdhNwAccessRule]* child resource under the same parent *[cmdhNetworkAccessRules]* resource when processing requests matching with the parent *[cmdhNetworkAccessRules]* resource's *targetNetwork*. Each Underlying Network name or the string 'default' shall appear at most once in any of the *targetNetwork* attributes of any of the provisioned *[cmdhNwAccessRule]* child resources under the same parent *[cmdhNetworkAccessRules]* resource.
minReqVolume	1	RW	Minimum amount of data that needs to be aggregated before any of the Underlying Networks matching with the *targetNetwork* attribute of this *[cmdhNwAccessRule]* resource can be used for forwarding information to other CSEs.

Attributes of *[cmdhNwAccessRule]*	Multiplicity	RW/ RO/ WO	Description
backOffParameters	1	RW	Parameters that define how usage of any of the Underlying Networks matching with the *targetNetwork* attribute of this *[cmdhNwAccessRule]* resource shall be handled when attempts to use such networks have failed. The *backOffParameters* attribute consists of 3 values: • A back-off time that defines how long a CSE needs to wait before attempting to use a specific Underlying Network again after a first failed attempt • A back-off time increment that defines by how much the back-off time shall be increased after each additional consecutive failed attempt to use the same Underlying Network without success • A maximum back-off time that defines the maximum wait time before attempting to use an Underlying Network again after previous failures.
otherConditions	0..1 (L)	RW	List of additional conditions that need to be fulfilled before any of the Underlying Networks matching with the *targetNetwork* attribute of this *[cmdhNwAccessRule]* resource can be used for forwarding information to other CSEs.
mgmtLink	1	RW	Link to an instance *allowedSchedule* of a *<schedule>* resource as defined in clause 9.6.9

D.12.8 Resource *cmdhBuffer*

The *[cmdhBuffer]* resource is used to define limits in usage of buffers for temporarily storing information that needs to be forwarded to other CSEs during processing of CMDH-related requests in a CSE. When an incoming request is processed by a CSE, it can only use buffers for temporary storage in compliance with the rules defined by the corresponding *[cmdhBuffer]* resource.

If a request cannot be processed in compliance with the rules defined in the corresponding *[cmdhBuffer]* resource, that request shall either be rejected in case it has not already been accepted by the CSE or it has to be purged. Error reporting on failed CMDH processing depends on error reporting parameters (TBD).

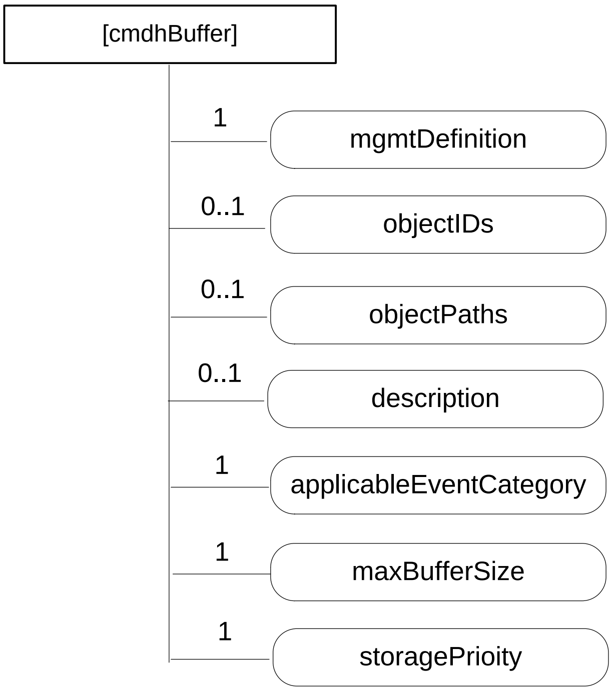


Figure D.12.8-1: Structure of *[cmdhBuffer]* resource

The *[cmdhBuffer]* resource shall contain attributes specified in table D.12.8-1.

Table D.12.8-1: Attributes of *[cmdhBuffer]* resource

Attributes of *[cmdhBuffer]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1 where this common attribute is described.
resourceID	1	WO	See clause 9.6.1 where this common attribute is described.
parentID	1	RO	See clause 9.6.1 where this common attribute is described.
expirationTime	1	RW	See clause 9.6.1 where this common attribute is described.
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1 where this common attribute is described.
creationTime	1	RO	See clause 9.6.1 where this common attribute is described.
lastModifiedTime	1	RO	See clause 9.6.1 where this common attribute is described.
labels	0..1	RO	See clause 9.6.1 where this common attribute is described
mgmtDefinition	1	WO	See clause 9.6.15. Has fixed value *"cmdhBuffer"*.
objectIDs	0..1	WO	See clause 9.6.15.
objectPaths	0..1	WO	See clause 9.6.15.
description	0..1	RW	See clause 9.6.15.
applicableEventCategory	1	RW	The *applicableEventCategory* attribute defines for which requests the limits contained in this *[cmdhBuffer]* resource shall be applied. The *applicableEventCategory* attribute is a list of zero or more Event Category values (**ec** parameter of a request), or the string 'default'. When an Event Category value appears in the *applicableEventCategory* attribute, the buffer usage limits defined inside this *[cmdhBuffer]* resource are applicable for requests associated with that specific Event Category (***ec***) value. When the string 'default' appears in the *applicableEventCategory* attribute, the buffer usage limits defined inside this *[cmdhBuffer]* resource are applicable for all requests whose associated Event Category value (**ec**) is not listed in the *applicableEventCategory* attribute of any other provisioned *[cmdhBuffer]* resource on the hosting CSE. A specific Event Category value (**ec**) shall appear at most once in any of the *applicableEventCategory* attributes of any of the provisioned *[cmdhBuffer]* resources on the hosting CSE. The string 'default' shall appear exactly once in any of the *applicableEventCategory* attributes of any of the provisioned *[cmdhBuffer]* resources on the hosting CSE.
maxBufferSize	1	RW	Maximum amount of memory that can be used for buffering requests matching with the *applicableEventCategory* attribute of this *[cmdhBuffer]* resource.
storagePriority	1	RW	Storage priority for data that is stored for buffering requests matching with the attribute of this *[cmdhBuffer]* resource. The storage priority defines the how to handle purging of buffered data when buffer memory is exhausted and buffered requests need to be purged. Buffered requests associated with a lower storage priority shall be purged before buffered requests with a higher storage priority. The range of storage priority is from 1 to 10.

Annex E (informative):
CSE Minimum Provisioning

The present clause defines the minimum set of resources instantiated in a CSE node with the scope to make it ready to provide services to entities that will register to.

For the purpose of the initial configuration two roles are identified:

- **superuser:** this role allows the full CSE control according to infrastructure provider policies. Only one superuser role is allowed per CSE;
- **user:** is the role associated to an AE that will register itself to Registrar CSE. More than one user roles are allowed per CSE. More than one applications can access to CSE with the same role.

Superuser role may be created with the following associated resources:

1) Definition or assignment of CSE-ID name that may be unique in the node hosting the CSE to be instantiated.

2) Creation of <CSEBase> resource with name equal to CSE-ID.

3) Creation of following child resources belonging to a tree with <CSEBase> as root:

   a) <accessControlPolicy> child resource enabling full access control for superuser's invoked operations to the tree resources. Subsequent created resources may have *accessControlPolicyIDs* attribute addressing this <accessControlPolicy> resource.

   b) <AE> child resource to be used as registered AE dedicated to superuser related activities.

Each user role may be created with the following associated resources:

1) Definition or assignment of an AE name that may be unique in the CSE.

2) Creation of <AE> child resource of <CSEBase> resource named as described in step 1, to be used as registered application dedicated to user related activities.

3) Creation of following child resources belonging to a tree with <AE> as root:

   a) <accessControlPolicy> resource enabling partial access control (e.g. these resources cannot be deleted be the user, superuser's resources can only be read by user) for user's invoked operations to the tree resources. <AE> resource can be updated with *accessControlPolicyIDs* attribute addressing <accessControlPolicy> resource.

The above described operations may be executed in the node in order provide the elements and the access control privileges required to provide the initial access to resource operations.

Same user can create more than one <AE> resources and other child resource types.

Once user role resource trees have been created the registered AE associated to <AE> resource (defined for a user role in step 2) is able to create its own <container> resource to store business logic application data that can be shared to other registered AEs in a controlled way acting on its own <accessControlPolicy> resource.

Annex F (informative): Interworking / Integration of non-oneM2M solutions and protocols

F.1 Introduction

Non-oneM2M solutions are currently installed and will continue to evolve and to be adopted in future for specific deployments. Some of these solution are the evolution of M2M that have a long history and significant mass installations (e.g. the PLC-related protocols commonly used in building and industrial automation), and are also significantly represented by proprietary solutions, especially in terms of semantic of the data model. The non-oneM2M solutions are potentially used for:

- Legacy deployment: such solutions can make use of both, proprietary or standard protocols; often proprietary data models and functionality are combined with the use of standard protocol.
- New system deployment that privilege the vertical optimization rather the horizontal aspects.
- Area network deployment for which native IP based oneM2M is perceived as not optimized respect to the used technology.

For those non-oneM2M solutions oneM2M needs to provide a means to enable:

- Mixed deployment that are partially oneM2M compliant and partially not, where the oneM2M System provides the solution to integrate multiple technologies (e.g. to add new technologies on top of old installations).
- Hybrid deployment that are still using non-oneM2M protocol (proprietary/standard) and want to use at the same time some of the oneM2M functionalities. A typical case is the exchange of heavy data traffic outside the CSE (e.g. for video surveillance), together with the use of CSE services for control and light traffic exchange.

F.2 Interworking with non-oneM2M solutions through specialized interworking applications

The solution is based on the use of specialized interworking Application Entities that are interfaced to the CSE via standard Mca reference points.

Such specialized applications are named Inter-working Proxy and are described in figure F.2-1.

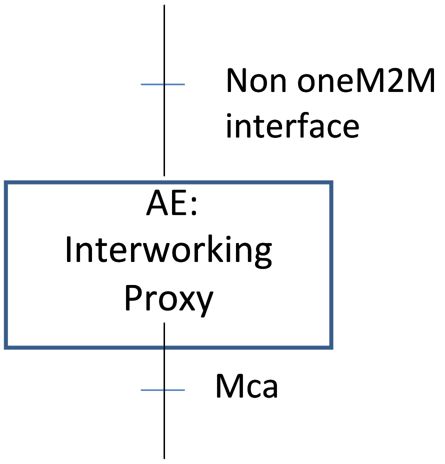


Figure F.2-1: Interworking Proxy

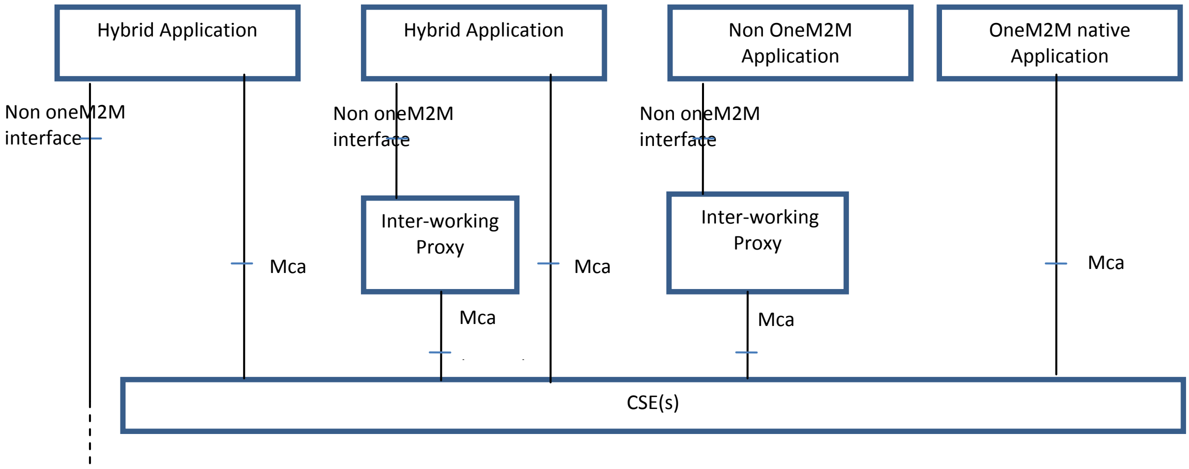
The Inter-working Proxy Application Entity (IPE) is characterized by the support of a non-oneM2M reference point, and by the capability of remapping the related data model to the oneM2M resources exposed via the Mca reference point.

This is typically supported via a full semantic inter-working of the data model used by the non oneM2M and a related protocol inter-working logic, and, depending on the complexity of the non oneM2M data model, can imply the definition of a complex set of resources built via the basic oneM2M ones, or a simple direct mapping of the communication via the containers.

The approach enable a unique solution for enabling communications among different protocols, catering for different level of inter-working including protocol inter-working, semantic information exchange, data sharing among the different solution and deployments.

And enables the offering additional values respect to what is today available via existing protocols and proprietary service exposures.

The following picture shows the typical scenarios supported by the oneM2M architecture in the context of inter-working. The combination of the different scenarios allows mixed deployments.



NOTE:	The additional option of an inter-working proxy embedded in the CSE as a module with an internal specified interface is under consideration.

Figure F.2-2: Scenarios Supported by oneM2M Architecture

These scenarios are applicable to the CSE with the AE as application dedicated node, in the application Service Node, in the Middle Node and in the infrastructure Node.

The following picture provides an example of the use of such capabilities an area network adopting specific protocols, e.g. Zigbee Telco Profile and Mbus using COSEM Data model.

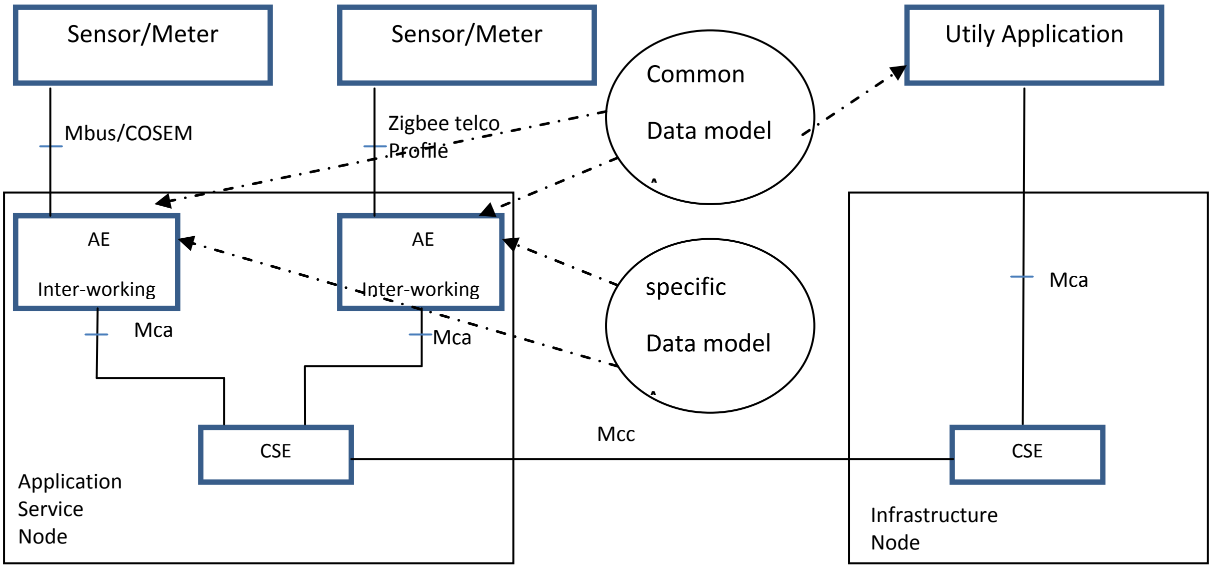


Figure F.2-3: TBD <provide title to the picture>

[F.2.a] Editor's Note: Figure title is needed.

There exist three variants of how interworking through an Inter-working Proxy Application Entity over Mca can be supported:

1) Interworking with full mapping of the semantic of the non-oneM2M data model to Mca.

   This is typically supported via a full semantic inter-working of the data model used by the non-oneM2M solution and the generic data model used in oneM2M (based on usage of containers) for exchanging application data. The IPE includes the related protocol inter-working logic.

   Depending on the complexity of the non-oneM2M data model, this can imply that the Inter-working Proxy Application Entity constructs a complex set of resources (built from the basic oneM2M resources) in the CSE. These resources are oneM2M representations of the non-oneM2M data model and are exposed by the IPE on Mca. They enable CSEs and AEs to access the entities in the non-oneM2M via the IPE.

   The benefit of this level of interworking is that it offers a unique solution for enabling communications among different protocols. The data model of the non-oneM2M solution determines its representation (the names, data types and structure of the containers) in the M2M System. It caters for different levels of inter-working including protocol inter-working, semantic information exchange, data sharing among the different solution and deployments. It enables offering additional values with respect to what is today available via existing protocols and proprietary service exposures.

   Note: With this level of interworking an M2M Application can access non-oneM2M solutions without the need to know the specific protocol encoding for these solutions. A drawback is that the IPE also potentially needs to interwork between a non-oneM2M security solution and oneM2M security. E.g. it needs to be the termination point of any non-oneM2M specific encryption.

2) Interworking using containers for transparent transport of encoded non-oneM2M data and commands via Mca.

   In this variant non-oneM2M data and commands are transparently packed by the Inter-working Proxy Application Entity into containers for usage by the CSEs and AEs.

   In this case the CSE or AE needs to know the specific protocol encoding rules of the non-oneM2M Solution to be able to en/de-code the content of the containers.

3) Interworking using a retargeting mechanism.

[F.2.b] Editor's Note: Retargeting mechanisms may also be mentioned, if and when defined.

F.3 Interworking versus integration of non-oneM2M solutions

Interworking:

With the approach given above - where specialized interworking applications (IPEs) allow to interact with any non-oneM2M system via the Mca interface - proprietary non-oneM2M solutions as well as non-oneM2M solutions that follow open standards can be interworked with the oneM2M System.

Integration:

When it is desired to make a certain type of non-oneM2M solution (e.g. some type of non-IP based Area Network) a permanent part of the deployed oneM2M Solution then the functionality of the Inter-working Proxy Application Entity can be integrated into the CSE of an Application Node. This is called "Integration" non-oneM2M solutions.

F.4 Entity-relation representation of non-IP based M2M Area Network

The following figure provides an entity-relation model that represents a non-IP based M2M area network as well as its relationship to an Interworking Proxy Application Entity (IPE).

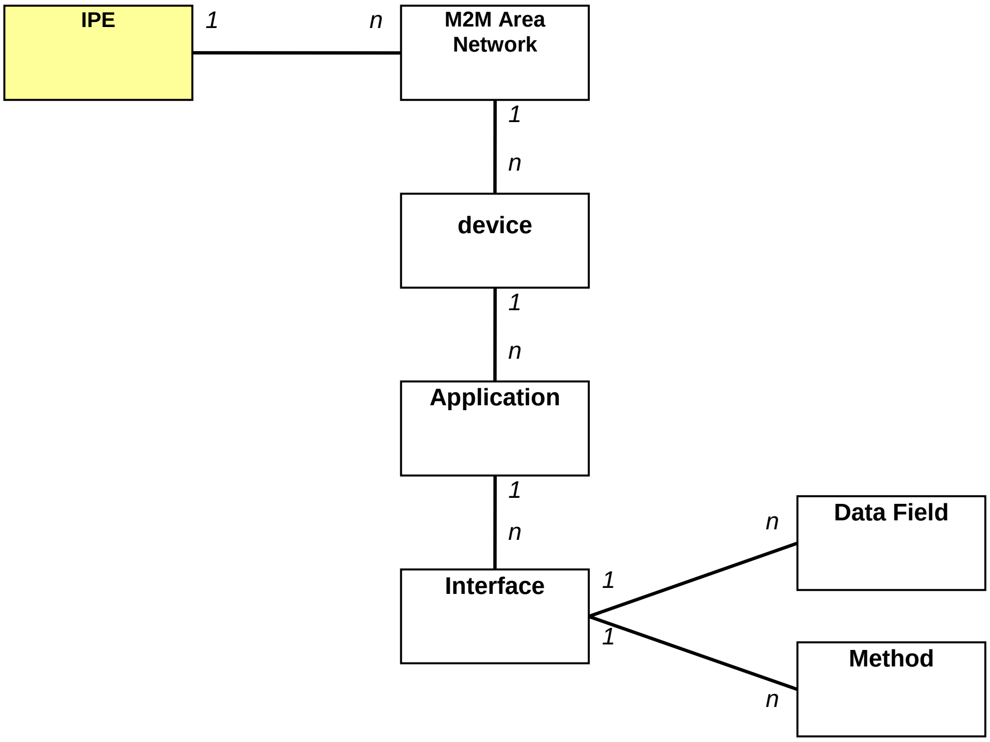


Figure F.4-1: Generic entity-relation diagram for an IPE and an M2M Area Network running legacy devices

This entity-relation diagram is e.g. applicable to the following M2M Area Networks:

- ZigBee.
- DLMS/COSEM.
- Zwave.

- BACnet.
- ANSI C12.
- mBus.

F.4.1 Responsibilities of Interworking Proxy Application Entity (IPE)

More specifically, the IPE is responsible to:

- create oneM2M resources representing the M2M Area Network structure (devices, their applications and interfaces) in the oneM2M Service Capability Layer, accessible via Mca;
- manage the oneM2M resources in case the M2M Area Network structure changes;
- discover the M2M Area Network structure and its changes automatically if this is supported by the technology of the M2M Area Network.

[F.4.1.a] Editor's Note: Mapping principles of the none-oneM2M information model into oneM2M resources is FFS.

Annex G (informative): List of M2M Services

This annex provides a list of M2M Services supported in this document as well as the list of associated roles mapped to each M2M Service.

Table G-1: M2M Services

M2M Service (name)	M2M-Serv-ID	Roles
Application and service management	01	001-Software management
Device management	02	002-Device Configuration 003-Device Diagnostics and Monitoring 004-Device Firmware Management 005-Device Topology
Location	03	006-Location
Data exchange	04	007-Basic data
Device onboarding	05	008-onboarding
Security	06	009-Security Administration

Use of M2M Service Subscription across M2M Service Provider domains is subject to M2M Service Providers agreement.

The following table provides an example of mapping of Service Roles to resource types and operations. Such a table is to be configured by the SP to allow for the validation of requests according to the service subscription.

Table G-2: Mapping of Service Roles to Resource Types/Allowed Operations

Service Role	Resource Type/Allowed Operations
001: Software management	• mgmtObj / CRUD • mgmtCmd / CRUD • software / CRUD
002: Device Configuration	• mgmtObj / CRUD • mgmtCmd / CRUD • deviceInfo / CRUD
003: Device Diagnostics and Monitoring	• mgmtObj / CRUD • mgmtCmd / CRUD • deviceInfo / CRUD • deviceCapability / CRUD
004: Device Firmware Management	• mgmtObj / CRUD • mgmtCmd / CRUD • firmware / CRUD
005: Device Topology	• mgmtObj / CRUD • mgmtCmd / CRUD • areaNwkInfo / CRUD • areaNwkDeviceInfo / CRUD
006: Location	• locationPolicy / CRUD • container / CRUD • subscription / CRUD
007: Basic data	• container / CRUD • subscription / CRUD
008: onboarding	• m2mServiceSubscription / CRUD • nodeInfo / CRUD
009: Security Administration	• accessControlPolicy / CRUD

Annex H (informative): Object Identifier Based M2M Device Identifier

H.1 Overview of Object Identifier

In M2M systems, it is required for devices to be distinguishable from one another through some kind of ID system. In other words, the ID which is allocated to the device is globally unique to ensure the proper operation of M2M systems, such as finding and connecting devices.

In relation to this requirement, the use of Object Identifiers may provide a convenient method to ensure the global uniqueness of M2M devices. The Object Identifier (OID) is an identification mechanism jointly developed by ITU-T and ISO/IEC which can be applied to objects, concepts, and all kinds of tangible or intangible things.

OID uses a hierarchical tree structure and is represented as a sequence of integer values, as shown in figure X. OID consists of several segments called arcs which provide placeholders for identification and description in the hierarchal tree. In the OID tree, the Root arc is unnamed and is represented by the forward slash (/) sign. The first arc represents the organization code and is used to manage and allocate its corresponding lower arc. The first arc can take the following values: itu-t (0), iso (1), and joint-iso-itu-t (2).

OID is hierarchically allocated, and the organization or the nation has the authority to define its lower arcs. For example, ITU-T can manage and allocate the lower arc below itu-t (0), and ISO can allocate the lower arc below iso (1). The general procedure regarding the use of OID is described in Recommendation ITU-T X.660 | ISO/IEC 9834-1 [i.27].

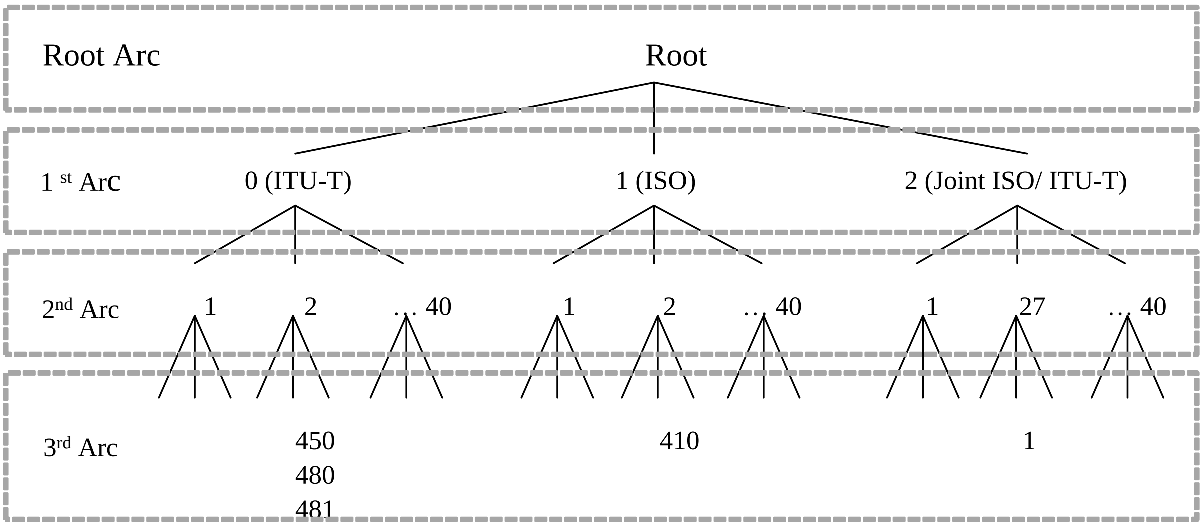


Figure H.1-1: International OID Tree

H.2 OID Based M2M Device Identifier

An M2M device shall be identified individually through a globally unique ID system. This clause explains how to allocate a globally unique ID to each M2M device by using the OID scheme. M2M device ID is an example which shows that OID can be applied to any M2M identifiers which need globally unique IDs.

The M2M device ID consists of a higher arc and a sequence of four arcs. It takes the form of {(higher arc) (x) (y) (z) (a)} as illustrated in figure Y. The higher arc is defined and managed according to the OID procedure. Each arc in the remaining sequence of four arcs represents the manufacturer ID, product model ID, serial number ID, and expanded ID, respectively.

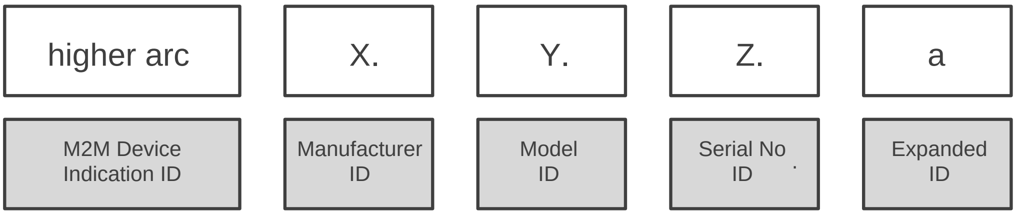


Figure H.2-1: M2M Device ID

H.2.1 M2M Device Indication ID - (higher arc)

The M2M Device Indication ID (higher arc) represents a globally unique identifier for the M2M device. The composition of the higher arc is variable and may be composed of several sub-arcs. The higher arc is assigned and managed by ITU-T/ISO.

H.2.2 Manufacturer ID - (x)

The 1st arc (x) among the sequential 4 arcs is used to identify the manufacturer which produces the M2M device. The first arc (x) is managed and allocated by the authority related with (higher arc).

H.2.3 Model ID - (y)

The 2nd arc (y) among the sequential 4 arcs identifies the device model produced by the manufacturer x. The second arc is managed and allocated by the manufacturer represented by the (x) arc.

H.2.4 Serial Number ID - (z)

The 3rd arc (z) among the sequential 4 arcs is for identifying the serial number of the device model y. The third arc is managed and allocated by the manufacturer represented by the (x) arc.

H.2.5 Expanded ID - (a)

The 4th arc (a) among the sequential 4 arcs is for identifying the legacy device which operates under the M2M device. The 4th arc for Expanded ID is allocated by the M2M device by adding a 4th arc to its device ID {(higher arc) (x) (y) (z)}. Therefore, the ID of legacy device which operates under the M2M device takes the form of {(higher arc) (x) (y) (z) (a)}. The fourth arc is managed and allocated by the M2M device.

H.3 Example of M2M device ID based on OID

Let us assume an M2M Device ID of {0 2 481 1 100 3030 10011}. The M2M device ID can be interpreted as follows:

- (0 2 481 1) in {0 2 481 1 100 3030 10011} - represents the M2M Device Indication ID (higher arc)
  - (0) in {0 2 481 1 100 3030 10011} - identifies the managing organization ITU-T
  - (2) in {0 2 481 1 100 3030 10011} - identifies "Administration"
  - (481) in {0 2 481 1 100 3030 10011} - identifies the data country code for Korea
  - (1) in {0 2 481 1 100 3030 10011} - identifies an M2M device
- (100) in {0 2 481 1 100 3030 10011} - identifies the device Manufacturer
- (3030) in {0 2 481 1 100 3030 10011} - identifies the device Model
- (10011) in {0 2 481 1 100 3030 10011} - identifies the device Serial number

History

This clause shall be the last one in the document and list the main phases (all additional information will be removed at the publication stage).

Publication history		
V1.1.1	<dd-Mmm-yyyy>	<Milestone>