ONEM2M
TECHNICAL SPECIFICATION

Document Number	TS-0032 V3.0.0
Document Name:	MAF and MEF Interface Specification
Date:	2019-02-22
Abstract:	This specification defines the reference points Mmaf and Mmef of oneM2M nodes (ADN, ASN, MN, IN) with the M2M Authentication Function (MAF) and the M2M Enrolment Function (MEF)

Template Version: 08 September 2015 (Dot not modify)

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

Contents

1 Scope

The present document specifies communication between the M2M Authentication Function (MAF) and MAF clients on the reference point Mmaf and between the M2M Enrolment Function (MEF) and MEF clients on the reference point Mmef.

2 References

2.1 Normative references

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

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

[1] oneM2M TS-0001: "Functional Architecture".

[2] oneM2M TS-0003: "Security Solutions".

[3] oneM2M TS-0004: "Service Layer Core Protocol Specification".

[4] oneM2M TS-0008: "CoAP Protocol Binding".

[5] oneM2M TS-0009: "HTTP Protocol Binding".

[6] oneM2M TS-0010: "MQTT Protocol Binding".

[7] oneM2M TS-0011: "Common Terminology".

[8] oneM2M TS-0020: "WebSocket Protocol Binding".

[9] oneM2M TS-0022: "Field Device Configuration".

2.2 Informative references

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

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

[i.1] oneM2M Drafting Rules.

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

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in oneM2M TS-0011 [7], oneM2M TS-0003 [2] and the following apply:

MAF Client: functionality for performing MAF procedures on behalf of an associated CSE or AE, or on behalf of CSE or AE(s) present on an associated Node

MAF interface: communication interface between a MAF and a MAF Client identified by reference point Mmaf

MEF Client: functionality for performing MEF procedures on behalf of an associated CSE or AE, or on behalf of CSE or AE(s) present on an associated Node, or an associated MAF

MEF interface: communication interface between a MEF and a MEF Client identified by reference point Mmef

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in oneM2M TS-0011 [7], oneM2M TS-0003 [2] and the following abbreviations apply:

ADN	Application Dedicated Node
AE	Application Entity
AE-ID	Application Entity Identifier
API	Application Programming Interface
ASN	Application Service Node
BBF	Broadband Forum
CDT	Common Data Types
CRUD	Create, Retrieve, Update, Delete (operation)
CSE	Common Services Entity
CSE-ID	Common Services Entity Identifier
DM	Device Management
DTLS	Datagram Transport Layer Security
FQDN	Fully Qualified Domain Name
HTTP	Hypertext Transfer Protocol
IN	Infrastructure Node
MAF	M2M Authentication Function
MEF	M2M Enrolment Function
MN	Middle Node
MQTT	Message Queue Telemetry Transport
MTE	M2M Trust Enabler
NP	Not Present
RSPF	Remote Security Provisioning Framework
RO	Read-Only
RW	Read-Write
SEC	Security
SP	Service Provider
SP-ID	Service Provider Identifier
SUID	Security Usage Identifier
TLS	Transport Layer Security
WO	Write-Only
XML	Extensible Markup Language

4 Conventions

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

5 General Description

5.1 MAF Interface

5.1.1 Introduction

The MAF Interface is a simple variant of the Mcc/Mca reference points specifying the interaction of MAF Clients with a M2M Authentication Function (MAF), acting on behalf of an *administrating stakeholder* such as an M2M SP or third party M2M Trust Enabler (MTE). The present document does not specify the operation and management of the MAF required to support these procedures.

A MAF Client interacts with the MAF on behalf of a Node (ADN, ASN, IN or MN), or a CSE or an AE.

Figure 5.1.1-1 defines the reference point Mmaf between MAF clients and a MAF.

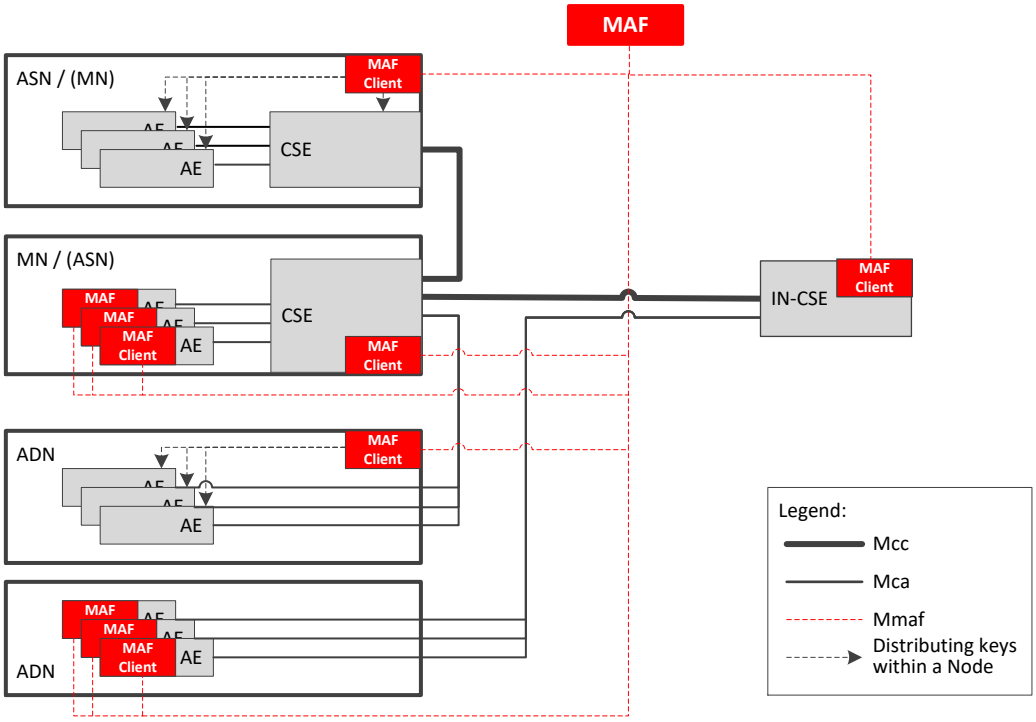

Figure 5.1.1-1: Reference Architecture for MAF

The administrating stakeholder authorizes the MAF's services to MAF clients, and oversees authorizing the distribution of symmetric keys. A MAF may provide its services on behalf of multiple administrating stakeholders. A MAF Client may be associated with multiple administrating stakeholders, each administrating the use of the MAF within a different scope.

NOTE 1: The administrating stakeholder could be an M2M SP administrating the registration and distribution of credentials used for SAEFs and ESPrim within the M2M SP's Domain.

NOTE 2: The administrating stakeholder could be an MTE administrating the registration and distribution of credentials for ESPrim and ESData to MAF Clients belonging to a particular Application Service Provider, where the MAF Clients could be distributed over multiple M2M SP domains.

The present document has no impact on the specifications in oneM2M TS-0001 [1] and oneM2M TS-0004 [3]. However, the MAF Interface uses much of the specification in oneM2M TS-0004 [3] and in particular allows use of the HTTP binding in oneM2M TS-0008 [4], the CoAP binding in oneM2M TS-0009 [5] and the WebSocket binding in oneM2M TS-0020 [8].

NOTE 3: The MQTT binding in oneM2M TS-0010 [6] is not suitable for the MAF Interface, because the MAF Interface assumes a TLS or DTLS connection from the MAF Client to the MAF – which is not possible using the MQTT binding.

The MAF Interface incorporates the following concepts from the Mcc/Mca reference points:

- The concept of operations acting on resources.
- The resource addressing from Mcc/Mca is used.
- The universal attributes and some common attributes of resources.

The MAF Interface differs from Mcc/Mca in the following ways:

- The MAF Client can only communicate directly with the MAF – there are no transited CSEs. Only Blocking Mode communication method is supported.
- None of the resource types applicable on Mcc/Mca are used:
  - Access control decisions use simple access control list for Retrieve access, and <*accessControlPolicy*> resources are not used for resources hosted by the MAF. A consequence of this is that the accessControlPolicyIDs attributes are not needed in the resources hosted by the MAF.
  - The <*subscription*> resource and NOTIFY operations are not supported.
  - There is no AE registration or CSE registration, but a similar process where a MAF Client creates a <*mafClientReg*> (MAF Client registration record) resource on the MAF.
  - There are no announced resources.

The hierarchy of resources hosted by a MAF shall be as follows:

- <*MAFBase*> resource type is the structural root for all the resources that are residing on a MAF. This resource is implicitly created by the MAF and uses the fixed resource name "maf" and contains following child resources:
  - <*mafClientReg*> resource. It confirms the MAF Client's registration to an administrating stakeholder, and can contain configuration information to be returned to the MAF Client.
  - <*symmKeyReg*> resources. It is created by the MAF Client, and contains symmetric keys for retrieval by another MAF Client.

5.1.2 MAF Interface Overview

This MAF Interface overview is based on the specification in clause 6 of oneM2M TS-0004 [3].

Identifiers such as M2M-SP-ID, AE-ID and CSE-ID as defined in 6.2.3 of [3] also apply to the MAF Interface. M2M Trust Enablers (MTEs) are identified using an M2M-SP-ID.

Resources are addressed as specified in clause 6.2.4 in [3].

Common data types applicable to the MAF Interface are inherited from clause 6.3 of [3]. However, for any parameters or elements which have assigned the enumerated data type m2m:resourceType, the applicable enumeration values are interpreted as specified in Table 5.1.2-4. This applies to the ***Resource Type*** primitive parameter, the common *resourceType* attribute, and the *@type* attribute of m2m:childResourceRef.

Table 5.1.2-1 and 5.1.2-2 list the request and response primitive parameters inherited from clauses 6.4.1 and 6.4.2 in [3], respectively; the data types of these parameters are unchanged. The ***From*** parameter shall include the MAF client ID which can be a Node-ID, AE-ID or CSE-ID, depending on whether the client acts on behalf of a node, AE or CSE.

Note that this is in contrast to primitives on the Mca and Mcc interface, where the ***From*** primitive parameter cannot include a Node-ID.

NOTE: All other optional request and response primitive parameters defined in clause 6.4.1 of [3] are not used on the MAF Interface.

Table 5.1.2-1: MAF Interface request primitive parameters

Parameter	Multiplicity	Notes
Operation	1	
To	1	
From	0..1	If not present, the MAF internally assigns ***From*** to be the identity of the Node, CSE or AE associated with the credential used for the MAF Handshake procedure.
Request Identifier	1	
Resource Type	0..1	values of m2m:resourceType interpreted as in Table 5.1.2-4
Content	0..1	
Result Content	0..1	

Table 5.1.2-2: MAF Interface response primitive parameters

Parameter	Multiplicity	Notes
Response Status Code	1	
Request Identifier	1	
Content	0..1	

Data types associated with resources applicable to the MAF Interface are defined in clause 7.

Table 5.1.2-3 lists the response status codes from clause 6.6 of [3] which are supported by the MAF Interface.

Table 5.1.2-3: Response status codes supported by the MAF Interface

Response status codes	Interpretation
2000	OK
2001	CREATED
2002	DELETED
2004	UPDATED
4000	BAD_REQUEST
4004	NOT_FOUND
4005	OPERATION_NOT_ALLOWED
4103	ACCESS_DENIED
5000	INTERNAL_SERVER_ERROR

Table 5.1.2-4: Interpretation of enumeration values of m2m:resourceType

Value	Interpretation	Note
1	MAFBase	
2	MEFBase	
3	mafClientReg	
4	mefClientReg	
5	symmKeyReg	
6	mefClientCmd	

The MIME media types defined on clause 6.7 of [3] shall be supported on the MAF interface. The notification related Media types vnd.onem2m-ntfy+json, vnd.onem2m-ntfy+cbor, vnd.onem2m-preq+xml do not apply to the MAF interface.

Virtual resources (clause 6.8 of [3]) are not supported by the MAF Interface.

5.2 MEF Interface

5.2.1 Introduction

The M2M Enrolment Function (MEF) is an essential part of the oneM2M Remote Security Provisioning architecture.

Clause 6.1.2.1 of oneM2M TS-0003 [2] defines the following three variants of Remote Security Provisioning Frameworks (RSPF):

- Pre-Provisioned Symmetric Key RSPF,
- Certificate-Based RSPF,
- GBA-based RSPF.

The MEF interface defined in the present specification applies to Pre-Provisioned Symmetric Key RSPF and Certificate-Based RSPF only. For interfaces and procedures applicable to GBA-based RSPF, see clause 8.3.2.3 of oneM2M TS-0003 [2].

When using Pre-Provisioned Symmetric Enrolee Key RSPF or Certificate-Based RSPF, the MEF serves a number of different use cases which are summarized as follows:

1) The MEF provisions an Enrolee to perform MAF Security Framework procedures with a MAF as defined in clause 8.8.2 of oneM2M TS-0003 [2].

2) The MEF provisions an Entity A and an Entity B to perform Security Association Establishment as defined in clauses 8.2.2.1 and 8.2.2.2 of oneM2M TS-0003 [2].

3) The MEF provisions an originator and a receiver of a primitive with credentials to enable End-to-End Security of Primitives (ESPRIM) with security credentials as specified in clause 8.4 of oneM2M TS-0003 [2].

4) The MEF provisions the source and target endpoints of End-to-End Security of Data (ESDATA) as specified in clause 8.5 of oneM2M TS-0003 [2].

The present document defines messages and procedures for the above listed MEF use cases.

NOTE 1: A MEF may also be implemented as a Device Management server using device management protocols such as OMA DM, OMA LwM2M and BBF TR-069. Such procedures are defined in oneM2M TS-0003 [2] and oneM2M TS-0022 [9].

Like the Mmaf Interface, the Mmef Interface is a simple variant of the Mcc/Mca reference points specifying the interaction of MEF Clients with a M2M Enrolment Function (MEF), managing symmetric keys on behalf of an *administrating stakeholder* such as an M2M SP or third party M2M Trust Enabler (MTE). The present document does not specify the operation and management of the MEF required to support these procedures.

A MEF Client interacts with the MEF on behalf of a Node (ADN, ASN, IN or MN), or a CSE or an AE for use case 1 and 2 in the above list. Figure 5.2.1-1 defines the reference point Mmef between MEF clients and a MEF, and between MEF and MAF.

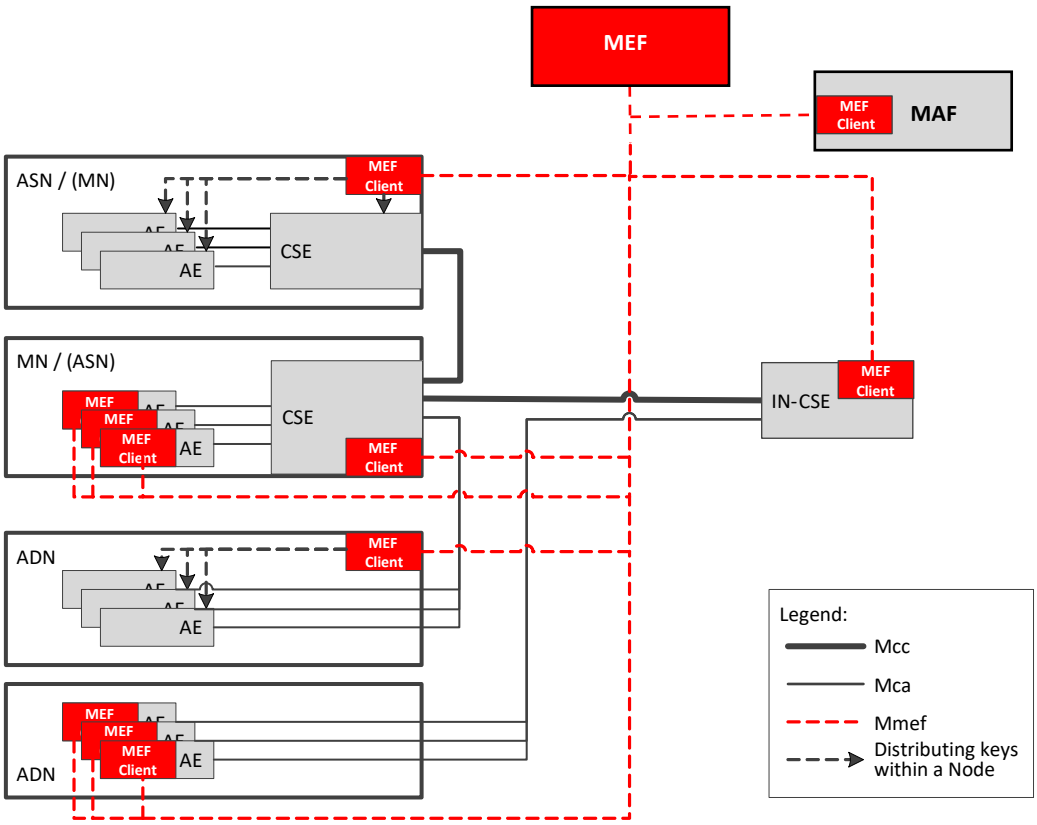

Figure 5.2.1-1: Reference Architecture for MEF

The administrating stakeholder authorizes the MEF's services to MEF clients. A MEF may provide its services on behalf of multiple administrating stakeholders. A MEF Client may be associated with multiple administrating stakeholders, each administrating the use of the MEF within a different scope.

NOTE 2: The administrating stakeholder could be an M2M SP administrating the registration and distribution of credentials used for SAEFs and ESPrim within the M2M SP's Domain.

NOTE 3: The administrating stakeholder could be an MTE administrating the registration and distribution of credentials for ESPrim and ESData to MEF Clients belonging to a particular Application Service Provider, where the MEF Clients could be distributed over multiple M2M SP domains.

The present document has no impact on the specifications in oneM2M TS-0001 [1] and oneM2M TS-0004 [3]. However, the MEF Interface uses much of the specification in oneM2M TS-0004 [3] and in particular allows use of the HTTP binding in oneM2M TS-0008 [4], the CoAP binding in oneM2M TS-0009 [5] and the WebSocket binding in oneM2M TS-0020 [8].

NOTE 4: The MQTT binding in oneM2M TS-0010 [6] is not suitable for the MEF Interface, because the MEF Interface assumes a TLS or DTLS connection from the MEF Client to the MEF – which is not possible using the MQTT binding

The MEF Interface incorporates the following concepts from the Mcc/Mca reference points:

1) The concept of operations acting on resources.

2) The resource addressing from Mcc/Mca is used.

3) The universal attributes and some common attributes of resources.

The MEF Interface differs from Mcc/Mca in the following ways:

4) The MEF Client can only communicate directly with the MEF - there are no transited CSEs. Only Blocking Mode communication method is supported.

5) None of the resource types applicable on Mcc/Mca are used:

- Access control decisions use simple access control list for Retrieve access, and <*accessControlPolicy*> resources are not used for resources hosted by the MEF. A consequence of this is that the accessControlPolicyIDs attributes are not needed in the resources hosted by the MEF.
- The <*subscription*> resource and NOTIFY operations are not supported.
- There is no AE registration or CSE registration, but a similar process where a MEF Client creates a <*mefClientReg*> (MEF Client registration record) resource on the MEF.
- There are no announced resources.

The hierarchy of resources hosted by a MEF shall be as follows:

6) <*MEFBase*> resource type is the structural root for all the resources that are residing on a MEF. This resource is implicitly created by the MEF and uses the fixed resource name "mef" and contains following child resources:

- <*mefClientReg*> resource. It confirms the MEF Client's registration to an administrating stakeholder, and can contain configuration information to be returned to the MEF Client.
- <*symmKeyReg*> resources. It is created by the MEF Client, and contains symmetric keys for retrieval by another MEF Client.

5.2.2 MEF Interface Overview

This MEF Interface overview is based on the specification in clause 6 of oneM2M TS-0004 [3].

Identifiers such as M2M-SP-ID, AE-ID and CSE-ID as defined in 6.2.3 of [3] also apply to the MEF Interface. M2M Trust Enablers (MTEs) are identified using an M2M-SP-ID.

Resources are addressed as specified in clause 6.2.4 in [3].

Common data types applicable to the MEF Interface are inherited from clause 6.3 of [3]. However, for any parameters or elements which have assigned the enumerated data type m2m:resourceType, the applicable enumeration values are interpreted as specified in Table 5.1.2-4. This applies to the ***Resource Type*** primitive parameter, the common *resourceType* attribute, and the *@type* attribute of m2m:childResourceRef.

Table 5.2.2-1 and 5.2.2-2 list the request and response primitive parameters inherited from clauses 6.4.1 and 6.4.2 in [3], respectively; the data types of these parameters are unchanged. The ***From*** parameter shall include the MEF client ID which can be a Node-ID, AE-ID or CSE-ID, depending on whether the client acts on behalf of a node, AE or CSE. Note that this is in contrast to primitives on the Mca and Mcc interface, where the ***From*** primitive parameter cannot include a Node-ID.

NOTE: All other optional request and response primitive parameters defined in clause 6.4.1 of [3] are not used on the MEF Interface.

Table 5.2.2-1: MEF Interface request primitive parameters

Parameter	Multiplicity	Notes
Operation	1	
To	1	
From	0..1	If not present, the MEF internally assigns ***From*** to be the identity of the Node, CSE or AE associated with the credential used for the MEF Handshake procedure.
Request Identifier	1	
Resource Type	0..1	values of m2m:resourceType interpreted as in Table 5.1.2-4
Content	0..1	
Result Content	0..1	

Table 5.2.2-2: MEF Interface response primitive parameters

Parameter	Multiplicity	Notes
Response Status Code	1	
Request Identifier	1	
Content	0..1	

Data types associated with resources applicable to the MEF Interface are defined in clause 7.

The response status codes listed in table 5.1.2-3 also apply to the MEF Interface.

The MIME media types defined on clause 6.7 of [3] shall be supported on the MEF interface. The notification related Media types vnd.onem2m-ntfy+json, vnd.onem2m-ntfy+cbor, vnd.onem2m-preq+xml do not apply to the MEF interface.

Virtual resources (clause 6.8 of [3]) are not supported by the MEF Interface.

6 Processing and Representation of Primitives

6.1 Common aspects of the MAF and MEF interface

This clause corresponds to the specification in clause 7 and 8 of oneM2M TS-0004 [3].

Both, MAF and MEF Interface request primitive formats conform to clause 7.2.1.1 [3], constrained to the CRUD operations, with request parameters listed in table 5.1.2-1 and table 5.2.2-1, respectively.

Both, MAF and MEF Interface response primitive formats conform to clause 7.2.1.2 [3], constrained to the CRUD operations, with response parameters listed in table 5.1.2-2 and table 5.2.2-2, respectively.

6.2 MAF Interface

The MAF Interface generic resource request procedure for originators and receivers conforms to clauses 7.2.2.1 and 7.2.2.2 of oneM2M TS-0004 [3], with the following clarification:

- The MAF Client acts as the originator, and the MAF acts as the receiver and resource hosting entity.
- The MAF Handshake procedure (clause 8.8.2.2 of oneM2M TS-0003 [2]) is used for mutual authentication of the MAF Client and MAF.
- The operation shall be one of the CRUD operations.
- The request and response parameters shall conform to table 5.1.2-1 and table 5.1.2-2.
- "Blocking Mode" communication method shall be used.
- The step Recv-6.3: "Check authorization of the Originator" is replaced by the authorization processes described in the MAF Interface resource-type specific procedures in clause 8.

The originator actions, receiver actions and Hosting CSE actions conform to clause 7.3 of [3], with clause 7.3.3.15 of [3] replaced by the authorization processes described in the MAF Interface resource-type specific procedures in clause 8.

The management common operations in clause 7.3.4 of [3] do not apply to the MAF Interface.

The resource-type-specification conventions apply to the specification in clause 8, but the remainder of clause 7.4 of [3] does not apply to the MAF Interface.

Clause 7.5.1 of [3] (regarding Notification) does not apply to the MAF Interface. Elements contained in the Content primitive parameter conform to clause 7.5.2 of [3].

The representation of MAF Interface primitives in data transfer conforms to clause 8. Clause 9 contains additional short names specific to both, the MAF and MEF Interfaces.

6.3 MEF Interface

The MEF Interface generic resource request procedure for originators and receivers conforms to clauses 7.2.2.1 and 7.2.2.2 of oneM2M TS-0004 [3] with the following clarification:

- The MEF Client acts as the originator, and the MEF acts as the receiver and resource hosting entity.
- The MEF Handshake procedure (clause 8.3.5.2.2 of oneM2M TS-0003 [2]) is used for mutual authentication of the MEF Client and MEF.
- The operation shall be one of the CRUD operations.
- The request and response parameters shall conform to table 5.2.2-1 and table 5.2.2-2.
- "Blocking Mode" communication method shall be used.
- The step Recv-6.3: "Check authorization of the Originator" is replaced by the authorization processes described in the MEF Interface resource-type specific procedures in clause 8.

The originator actions, receiver actions and Hosting CSE actions conform to clause 7.3 of [3], with clause 7.3.3.15 of [3] replaced by the authorization processes described in the MEF Interface resource-type specific procedures in clause 8.

The management common operations in clause 7.3.4 of [3] do not apply to the MEF Interface.

The resource-type-specification conventions apply to the specification in clause 8, but the remainder of clause 7.4 of [3] does not apply to the MEF Interface.

Clause 7.5.1 of [3] (regarding Notification) does not apply to the MEF Interface. Elements contained in the Content primitive parameter conform to clause 7.5.2 of [3].

The representation of MEF Interface primitives in data transfer conforms to clause 8. Clause 9 contains additional short names specific to the both, the MAF and MEF Interfaces.

7 Resource types definitions

7.1 Namespaces used for resource and data types

Representations of resources applicable to the MAF and MEF Interfaces employ the namespace identifier "sec:" for global XML elements associated with a resource type. Data types of the attributes and complex-type elements of these resource types may use any of the name space identifiers listed in table 7.1-1

Any data types of XML elements defined for use in present document shall be one of name spaces in table 7.1-1.

Table 7.1-1: Namespaces applicable to resource types defined in the present document

Name space	prefix	Name space definition	Types defined in
oneM2M Security	sec:	http://www.onem2m.org/xml/securityProtocols	the present document and TS-0003 [2]
oneM2M protocol CDT	m2m:	http://www.onem2m.org/xml/protocol	TS-0004 [3]
Device Configuration	dcfg:	http://www.onem2m.org/xml/deviceConfig	TS-0022 [9]

7.2 Resource Type <*MAFBase*>

The <*MAFBase*> resource shall represent a MAF.

The <*MAFBase*> resource shall contain the child resources specified in table 7.2-1.

Table 7.2-1: Child resources of <*MAFBase*> resource

Child Resources of <*MAFBase*>	Child Resource Type	Multiplicity	Description
[variable]	<*mafClientReg*>	0..n	See clause 7.3
[variable]	<*symmKeyReg*>	0..n	See clause 7.4

The <*MAFBase*> resource shall contain the attributes specified in table 7.2-2.

Table 7.2-2: Attributes of <*MAFBase*> resource

Attributes of <*MAFBase*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1.3 of [1].
resourceID	1	RO	See clause 9.6.1.3 of [1].
resourceName	1	RO	See clause 9.6.1.3 of [1]. Shall be fixed to "maf".
creationTime	1	RO	See clause 9.6.1.3 of [1].
labels	1	RO	See clause 9.6.1.3 of [1].

7.3 Resource Type <*MEFBase*>

The <*MEFBase*> resource shall represent a MEF.

The <*MEFBase*> resource shall contain the child resources specified in table 7.3-1.

Table 7.3-1: Child resources of <*MEFBase*> resource

Child Resources of <*MEFBase*>	Child Resource Type	Multiplicity	Description
[variable]	<*mefClientReg*>	0..n	See clause 7.4
[variable]	<*symmKeyReg*>	0..n	See clause 7.6

The <*MEFBase*> resource shall contain the attributes specified in table 7.3-2.

Table 7.3-2: Attributes of <*MEFBase*> resource

Attributes of <*MEFBase*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1.3 of [1].
resourceID	1	RO	See clause 9.6.1.3 of [1].
resourceName	1	RO	See clause 9.6.1.3 of [1]. Shall be fixed to "mef".
creationTime	1	RO	See clause 9.6.1.3 of [1].
labels	1	RO	See clause 9.6.1.3 of [1].

7.4 Resource Type <*mafClientReg*>

The <*mafClientReg*> resource shall represent a MAF Client enrolled with an M2M SP or M2M Trust Enabler (MTE).

NOTE: A single MAF Client can be enrolled with at most one M2M SP and any number of MTEs (typically enabling end-to-end security to MAF Clients outside the MAF Client's M2M SP's domain). Consequently, a single MAF Client can be associated with multiple <*mafClientReg*> resources on multiple MAFs. It is also possible that a single MAF Client can be associated with multiple <*mafClientReg*> resources on a single MAF acting on behalf of multiple administrating stakeholders.

The <*mafClientReg*> resource shall contain no child resources.

The <*mafClientReg*> resource shall contain the attributes specified in table 7.4-1.

Table 7.4-1: Attributes of <*mafClientReg*> resource

Attributes of <*mafClientReg*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1.3 of [1].
resourceID	1	RO	See clause 9.6.1.3 of [1].
resourceName	1	RO	See clause 9.6.1.3 of [1].
parentID	1	RO	See clause 9.6.1.3 of [1].
creationTime	1	RO	See clause 9.6.1.3 of [1].
labels	1	RW	See clause 9.6.1.3 of [1].
expirationTime	1	WO	See clause 9.6.1.3 of [1].
creator	1	WO	This attribute shall include the identifier of the MAF client which has created this resource.
adminFQDN	1	WO	FQDN of the M2M SP or MTE who is the administrating stakeholder of this enrolment.
assignedSymmKeyIID	0..1	RO	When the MAF Client uses a symmetric key to authenticate to the MAF, then the MAF may use this attribute to provide a symmetry key identifier within the domain of the MAF. Assigned by the MAF.

7.5 Resource Type <*mefClientReg*>

The <*mefClientReg*> resource shall represent a MEF Client enrolled with an M2M SP or M2M Trust Enabler (MTE).

NOTE: A single MEF Client can be enrolled with at most one M2M SP and any number of MTEs (typically enabling end-to-end security to MEF Clients outside the MEF Client's M2M SP's domain). Consequently, a single MEF Client can be associated with multiple <*mefClientReg*> resources on multiple MEFs. It is also possible that a single MEF Client can be associated with multiple <*mefClientReg*> resources on a single MEF acting on behalf of multiple administrating stakeholders.

The <*mefClientReg*> resource shall contain no child resources.

The <*mefClientReg*> resource shall contain the child resources specified in table 7.5-1.

Table 7.5-1: Child resources of <*mefClientReg*> resource

Child Resources of <*mefClientReg*>	Child Resource Type	Multiplicity	Description
"*cmd*"	<*mefClientCmd*>	1	See clause 7.7

The <*mefClientReg*> resource shall contain the attributes specified in table 7.5-2.

Table 7.5-2: Attributes of <*mefClientReg*> resource

Attributes of <*mefClientReg*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1.3 of [1].
resourceID	1	RO	See clause 9.6.1.3 of [1].
resourceName	1	RO	See clause 9.6.1.3 of [1].
parentID	1	RO	See clause 9.6.1.3 of [1].
creationTime	1	RO	See clause 9.6.1.3 of [1].
labels	1	RW	See clause 9.6.1.3 of [1].
expirationTime	1	WO	See clause 9.6.1.3 of [1].
creator	1	WO	This attribute shall include the identifier of the MEF client which has created this resource.
adminFQDN	1	WO	FQDN of the M2M SP or MTE who is the administrating stakeholder of this enrolment.
assignedSymmKeyIID	0..1	RO	When the MEF Client uses a symmetric key to authenticate to the MEF, then the MEF may use this attribute to provide a symmetry key identifier within the domain of the MEF. Assigned by the MEF.
sourceIDs	0..1	RW	List of AE-IDs and CSE-IDs associated with a MEF client acting on behalf of a Node. This attribute shall be supplied if the *creator* attribute includes a Node-ID

7.6 Resource Type <*symmKeyReg*>

The <*symmKeyReg*> resource shall represent a symmetric key that a source MAF Client or a source MEF Client has established with the MAF or MEF, respectively, for distributing to authorized Target MAF or MEF Clients and/or another MAF or MEF. The MAF or MEF Client provides a list of authorized Targets when the resource is created - the present document does not specify how the MAF or MEF associates the list with the resource. The MAF or MEF, in coordination with the identified administrating stakeholder (M2M SP or MTE), can modify the list of authorized Targets and the *expirationTime*.

The <*symmKeyReg*> resource shall contain no child resources.

The <*symmKeyReg*> resource shall contain the attributes specified in table 7.6-1.

Table 7.6-1: Attributes of <*symmKeyReg*> resource

Attributes of <*symmKeyReg*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1.3 of [1].
resourceID	1	RO	See clause 9.6.1.3 of [1].
resourceName	1	RO	See clause 9.6.1.3 of [1]. This value is used as the relative part of the identifier for the symmetric key in security protocols.
parentID	1	RO	See clause 9.6.1.3 of [1].
creationTime	1	RO	See clause 9.6.1.3 of [1].
labels	0..1	RW	See clause 9.6.1.3 of [1].
expirationTime	1	WO	See clause 9.6.1.3 of [1].
creator	1	RO	See clause 9.6.1.3 of [1].
adminFQDN	1	WO	FQDN of the administrating stakeholder (M2M SP or MTE) associated with this enrolment.
SUID	1	WO	An SUID constraining the use of the symmetric key associated with this resource.
targetIDs	1 (L)	RW	List of AE-ID(s) and/or CSE-ID(s) and/or and/or Node-ID(s) identifying the AE(s) and/or CSE(s) and/or Node(s) authorized to retrieve the resource. Only the creator and administrating stakeholder (identified by adminFQDN) are authorized to access this attribute.
keyValue	1	WO	The value of the key to be provided to the identifier targets. May be provided in the Create request or derived by the MAF or MEF Client and MAF or MEF from the TLS handshake parameters.

7.7 Resource Type <*mefClientCmd*>

A <*mefClientCmd*> resource includes instructions for the MEF client associated with the parent <*mefClientReg*> resource to be executed.

The <*mefClientCmd*> resource shall contain no child resources.

The <*mefClientCmd*> resource shall contain the attributes specified in table 7.7-1.

Table 7.7-1: Attributes of <*mefClientCmd*> resource

Attributes of <*mefClientCmd*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1.3 of [1].
resourceID	1	RO	See clause 9.6.1.3 of [1].
resourceName	1	RO	See clause 9.6.1.3 of [1].
parentID	1	RO	See clause 9.6.1.3 of [1].
creationTime	1	RO	See clause 9.6.1.3 of [1].
labels	1	RW	See clause 9.6.1.3 of [1].
expirationTime	1	WO	See clause 9.6.1.3 of [1].
cmdID	1	RW	This attribute shall include a MEF-assigned identifier of a command issued by the MEF. See clause 8.3.9.1 of [2] for further details.
cmdDescription	1	RO	This attribute provides the description of a command issued by the MEF to be executed by the MEF client. See clause 8.3.9.5 of [2] for further details.
cmdStatusCode	1	RW	This attribute shall be used forthe status of the command issued by the MEF. See clause 8.3.9.6 of [2] for further details.

8 Resource-type specific procedures and definitions

8.1 Resource Type <*MAFBase*>

8.1.1 Introduction

A <*MAFBase*> resource shall represent a MAF. This <*MAFBase*> resource shall be the root for all the resources that are residing on the MAF.

Table 8.1.1-1: Data Type Definition of <*MAFBase*>

Data Type ID	File Name	Note
MAFBase	SEC-MAFBase-v2_1_0.xsd	

The <*MAFBase*> resource has no resource-specific attributes.

Table 8.1.1-2: Child resources of <MAFBase> resource

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<mafClientReg>	[variable]	0..n	Clause 7.4
<symmKeyReg>	[variable]	0..n	Clause 7.6

8.1.2 <*MAFBase*> resource specific procedures on CRUD operations

8.1.2.1 Create

Originator:

The <*MAFBase*> resource shall not be created via API.

Receiver:

Primitive specific operation on Recv-1.0 "Check the syntax of received message":

1) If the request is received, the MAF shall execute the following steps in order:

   a) "Create an unsuccessful Response primitive" with the ***Response Status Code*** indicating "OPERATION_NOT_ALLOWED" error.

   b) "Send the Response primitive".

8.1.2.2 Retrieve

Originator:

No change from the generic procedures in clause 7.2.2.1 of [3] with clarifications discussed in clauses 5.1 and 6.

Receiver:

Same as the generic procedures in clause 7.2.2.2 of [3] with clarifications discussed in clauses 5.1 and 6, and performing the following step in the place of step Recv-6.3: "Check authorization of the Originator":

The Receiver shall allow all Originator's to retrieve this resource.

8.1.2.3 Update

Originator:

The <*MAFBase*> resource shall not be updated via API.

Receiver:

Primitive specific operation on Recv-1.0 "Check the syntax of received message":

1) If the request is received, the MAF shall execute the following steps in order:

   a) "Create an unsuccessful Response primitive" with the Response Status Code indicating "OPERATION_NOT_ALLOWED" error.

   b) "Send the Response primitive".

8.1.2.4 Delete

Originator:

The <MAFBase> resource shall not be DELETEed via API.

Receiver:

Primitive specific operation on Recv-1.0 "Check the syntax of received message":

1) If the request is received, the MAF shall execute the following steps in order:

   a) "Create an unsuccessful Response primitive" with the ***Response Status Code*** indicating "OPERATION_NOT_ALLOWED" error.

   b) "Send the Response primitive".

8.2 Resource Type <MEFBase>

8.2.1 Introduction

A <*MEFBase*> resource shall represent a MEF. This <*MEFBase*> resource shall be the root for all the resources that are residing on the MEF.

Table 8.2.1-1: Data Type Definition of <*MEFBase*>

Data Type ID	File Name	Note
MEFBase	SEC-MEFBase-v2_1_0.xsd	

The <*MEFBase*> resource has no resource-specific attributes.

Table 8.2.1-2: Child resources of <*MEFBase*> resource

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<*mefClientReg*>	[variable]	0..n	Clause 7.5
<*symmKeyReg*>	[variable]	0..n	Clause 7.6

8.2.2 <*MEFBase*> resource specific procedures on CRUD operations

8.2.2.1 Create

Originator:

The <*MEFBase*> resource shall not be created via API.

Receiver:

Primitive specific operation on Recv-1.0 "Check the syntax of received message":

1) If the request is received, the MEF shall execute the following steps in order:

   a) "Create an unsuccessful Response primitive" with the ***Response Status Code*** indicating "OPERATION_NOT_ALLOWED" error\

   b) "Send the Response primitive".

8.2.2.2 Retrieve

Originator:

No change from the generic procedures in clause 7.2.2.1 of [3] with clarifications discussed in clauses 5.2 and 6.

Receiver:

Same as the generic procedures in clause 7.2.2.2 of [3] with clarifications discussed in clauses 5.2 and 6, and performing the following step in the place of step Recv-6.3: "Check authorization of the Originator":

1) The Receiver shall allow all Originator's to retrieve this resource.

8.2.2.3 Update

Originator:

The <*MEFBase*> resource shall not be updated via API.

Receiver:

Primitive specific operation on Recv-1.0 "Check the syntax of received message":

1) If the request is received, the MEF shall execute the following steps in order:

   a) "Create an unsuccessful Response primitive" with the Response Status Code indicating "OPERATION_NOT_ALLOWED" error.

   b) "Send the Response primitive".

8.2.2.4 Delete

Originator:

The <MEFBase> resource shall not be DELETEed via API.

Receiver:

Primitive specific operation on Recv-1.0 "Check the syntax of received message":

1) If the request is received, the MEF shall execute the following steps in order:

   a) "Create an unsuccessful Response primitive" with the ***Response Status Code*** indicating "OPERATION_NOT_ALLOWED" error.

   b) "Send the Response primitive".

8.3 Resource Type <*mafClientReg*>

8.3.1 Introduction

A <*mafClientReg*> resource shall represent a MAF Client enrolled with the MAF on behalf of an M2M Service Provider or M2M Trust Enabler. A <*mafClientReg*> resource shall be a child resource of the MAF's <*MAFBase*> resource.

Table 8.3.1-1: Data Type Definition of <*mafClientReg*>

Data Type ID	File Name	Note
mafClientReg	SEC-mafClientReg-v2_1_0.xsd	

Table 8.3.1-2: Universal/Common Attributes of <*mafClientReg*> resource

Attribute Name	Request Optionality	
	Create	Update
@resourceName	NP	NP
resourceType	NP	NP
resourceID	NP	NP
parentID	NP	NP
creationTime	NP	NP
labels	O	O
expirationTime	M	M
creator	NP	NP

Table 8.3.1-3: Resource Specific Attributes of <*mafClientReg*> resource

Attribute Name	Request Optionality		Data Type	Default Value and Constraints
	Create	Update		
adminFQDN	M	NP	xs:anyURI	No default
assignedSymmKeyID	NP	NP	sec:credentialID	No default

The <*mafClientReg*> resource shall contain no child resources.

8.3.2 <*mafClientReg*> resource specific procedures on CRUD operations

8.3.2.1 Create

This procedure is denoted *MAF Client Registration* in clause 8.8.2.3 of oneM2M TS-0003 [2]. The ***To*** parameter of the <*mafClientReg*> create request primitive includes the MAF-FQDN and the character "–" (dash) as a shorthand notation for the name of the <*MAFBase*> resource:

//{MAF-FQDN}/–/

EXAMPLE: //maf123.mafprovider.org/–/

The MAF-FQDN represents a globally unique identifier of a MAF (aka. MAF ID).

The ***From*** parameter of the <*mafClientReg*> create request primitive shall be left empty if the MAF client does not have a MAF Client ID assigned yet. If the MAF client interfaces with the MAF on behalf of the node (see clause 5.1.1), the Node-ID of the respective ADN, ASN, MN or IN shall serve as MAF Client ID.

Originator:

No change from the generic procedures in clause 7.2.2.1 of [3] with clarifications discussed in clauses 5.1.2 and 6.2, and with following differences.

In step Orig-6.0: "Process Response primitive", if the Originator used a symmetric key to authenticate to the MAF, and the <*mafClientReg*> resource in the response contained an *assignedSymmKeyID* attribute then the originator shall use the *assignedSymmKeyID* attribute to identify this symmetric key when it is subsequently used in authenticating to the MAF.

Receiver:

Same as the generic procedures in clause 7.2.2.2 of [3] with clarifications discussed in clauses 5.1.2 and 6.2, and with following differences.

The Receiver shall perform the following steps in order in the place of Recv-6.3: "Check authorization of the Originator":

1) The Receiver shall determine if the Originator is authorized to register with the administrating stakeholder (M2M SP or MTE) identified by *adminFQDN* attribute. The present document does not specify how the Receiver makes this determination:

    a) If the Originator is not authorized, then the Receiver shall execute the following steps in order:

        i) "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS_DENIED" error.

        ii) "Send the Response primitive".

    b) If the Originator is authorized, then the Receiver shall allow the request.

The Receiver shall perform the following steps in order as part of "Create the resource" (clause 7.3.3.5 of [3]) during Step Recv-6.5: "Create/Update/Retrieve/Delete/Notify operation is performed":

2) If the Originator authenticated using symmetric key with a key identifier which does not use the Receiver's FQDN, then:

    a) The Receiver shall assign a symmetric key identifier with the Receiver's FQDN and with relative part which is unique within the scope of symmetric key identifiers issued by the Receiver. The Receiver shall associate this symmetric key identifier with the symmetric key used for authenticating the Originator.

    b) The Receiver shall set the *assignedSymmKeyID* attribute to be the Credential-ID formed from the assigned symmetric key identifier as specified in clause 10.4 of [2].

3) If the Originator authenticated using a symmetric key with a key identifier which does not use the Receiver's FQDN, or if the Originator authenticated using a certificate, then the Receiver shall not include an *assignedSymmKeyID* attribute in the created resource.

4) The Receiver shall assign the *creator* attribute to an AE-ID or CSE-ID or Node-ID on instructions from the administrating stakeholder. The present document does not specify any details of how the AE-ID or CSE-ID or Node-ID is determined.

8.3.2.2 Retrieve

This procedure is denoted *MAF Client Configuration Retrieval* in clause 8.8.2.4 of oneM2M TS-0003 [2]. This procedure is used to retrieve the <*mafClientReg*> resource.

Originator:

No change from the generic procedures in clause 7.2.2.1 of [3] with clarifications discussed in clauses 5.1.2 and 6.2.

Receiver:

Same as the generic procedures in clause 7.2.2.2 of [3] with clarifications discussed in clauses 5.1.2 and 6.2, performing the following steps in order in the place of Recv-6.3: "Check authorization of the Originator":

1) The Receiver shall determine if the Originator is authorized by checking if the Originator is the creator of the resource:

    a) If the Originator is not authorized, then the Receiver shall execute the following steps in order:

        i) "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS_DENIED" error.

        ii) "Send the Response primitive".

    b) If the Originator is authorized, then the Receiver shall allow the request.

8.3.2.3 Update

This procedure is denoted *MAF Client Configuration Update* in clause 8.8.2.5 of oneM2M TS-0003 [2]. This procedure is used to update attributes of the <*mafClientReg*> resource, such as e.g. labels, expiration time.

Originator:

The <*mafClientReg*> resource shall not be updated by a MAF client via API.

Receiver:

Same as the generic procedures in clause 7.2.2.2 of [3] with clarifications discussed in clauses 5.1.2 and 6.2, and with the following differences:

The Receiver shall perform the following step in the place of Recv-6.3: "Check authorization of the Originator":

1) The Receiver shall determine if the Originator is authorized by checking if the Originator is the creator of the resource:
    a) If the Originator is not authorized, then the Receiver shall execute the following steps in order:
        i) "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS_DENIED" error.
        ii) "Send the Response primitive".
    b) If the Originator is authorized, then the Receiver shall allow the request.

The Receiver shall perform the following step as part of "Update the resource" (clause 7.3.3.7 of [3])" during Step Recv-6.5: "Create/Update/Retrieve/Delete/Notify operation is performed":

2) If the Originator was the Creator of the resource, then the Receiver shall perform steps 2 and 3 in clause 8.3.2.1.

8.3.2.4 Delete

This procedure is denoted *MAF Client De-Registration* in clause 8.8.2.6 of oneM2M TS-0003 [2]. This procedure enables the MAF client to delete its own <*mafClientReg*> resource on a MAF.

Originator:

No change from the generic procedures in clause 7.2.2.1 of [3] with clarifications discussed in clauses 5.1.2 and 6.2.

Receiver:

Same as the generic procedures in clause 7.2.2.2 of [3] with clarifications discussed in clauses 5.1.2 and 6.2, performing the following step in the place of Recv-6.3: "Check authorization of the Originator":

1) The Receiver shall determine if the Originator is authorized by checking if the Originator is the creator of the resource:
    a) If the Originator is not authorized, then the Receiver shall execute the following steps in order:
        i) "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS_DENIED" error.
        ii) "Send the Response primitive".
    b) If the Originator is authorized, then the Receiver shall allow the request.

8.4 Resource Type <*mefClientReg*>

8.4.1 Introduction

A <*mefClientReg*> resource shall represent a MEF Client enrolled with the MEF on behalf of an M2M Service Provider or M2M Trust Enabler. A <*mefClientReg*> resource shall be a child resource of the MEF's <*MEFBase*> resource.

Table 8.4.1-1: Data Type Definition of <*mefClientReg*>

Data Type ID	File Name	Note
mefClientReg	SEC-mefClientReg-v2_1_0.xsd	

Table 8.4.1-2: Universal/Common Attributes of <*mefClientReg*> resource

Attribute Name	Request Optionality	
	Create	Update
@resourceName	NP	NP
resourceType	NP	NP
resourceID	NP	NP
parentID	NP	NP
creationTime	NP	NP
labels	O	O
expirationTime	M	M
creator	NP	NP

Table 8.4.1-3: Resource Specific Attributes of <*mefClientReg*> resource

Attribute Name	Request Optionality		Data Type	Default Value and Constraints
	Create	Update		
adminFQDN	M	NP	xs:anyURI	No default
assignedSymmKeyID	NP	NP	sec:credentialID	No default
sourceIDs	O	NP	List of m2m:ID	No default

Table 8.4.1-4: Child resources of <*mefClientReg*> resource

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<*mefClientCmd*>	”cmd”	1	Clause 7.7

8.4.2 <*mefClientReg*> resource specific procedures on CRUD operations

8.4.2.1 Create

This procedure is denoted *MEF Client Registration* in clause 8.3.5.2.3 of oneM2M TS-0003 [2]. The ***To*** parameter of the <*mefClientReg*> create request primitive includes the MEF-FQDN and the character "–" (dash) as a shorthand notation for the name of the <*MEFBase*> resource:

//{MEF-FQDN}/–/

EXAMPLE: //mef123.mefprovider.org/–/

The MEF-FQDN represents a globally unique identifier of a MEF (aka. MEF ID).

The ***From*** parameter of the <*mefClientReg*> create request primitive shall be left empty if the MEF client does not have a MEF Client ID assigned yet. If the MEF client interfaces with the MEF on behalf of the node (see clause 5.2.1), the Node-ID of the respective ADN, ASN, MN or IN shall serve as MEF Client ID.

Originator:

No change from the generic procedures in clause 7.2.2.1 of [3] with clarifications discussed in clauses 5.2.2 and 6.3, and with following differences:

In step Orig-6.0: "Process Response primitive", if the Originator used a symmetric key to authenticate to the MEF, and the <*mefClientReg*> resource in the response contained an *assignedSymmKeyID* then the originator shall use the *assignedSymmKeyID* to identify this symmetric key when it is subsequently used in authenticating to the MEF.

Receiver:

Same as the generic procedures in clause 7.2.2.2 of [3] with clarifications discussed in clauses 5.2.2 and 6.3, and with following differences:

The Receiver shall perform the following steps in order in the place of Recv-6.3: "Check authorization of the Originator":

1) The Receiver shall determine if the Originator is authorized to register with the administrating stakeholder (M2M SP or MTE) identified by *fqdn* attribute. The present document does not specify how the Receiver makes this determination:

    a) If the Originator is not authorized, then the Receiver shall execute the following steps in order:

        i) "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS_DENIED" error.

        ii) "Send the Response primitive".

    b) If the Originator is authorized, then the Receiver shall allow the request.

The Receiver shall perform the following steps in order as part of "Create the resource" (clause 7.3.3.5 of [3]) during Step Recv-6.5: "Create/Update/Retrieve/Delete/Notify operation is performed":

2) If the Originator authenticated using symmetric key with a key identifier which does not use the Receiver's FQDN, then:

    a) The Receiver shall assign a symmetric key identifier with the Receiver's FQDN and with relative part which is unique within the scope of symmetric key identifiers issued by the Receiver. The Receiver shall associate this symmetric key identifier with the symmetric key used for authenticating the Originator.

    b) The Receiver shall set the *assignedSymmKeyID* attribute to be the Credential-ID formed from the assigned symmetric key identifier as specified in clause 10.4 of [2].

3) If the Originator authenticated using a symmetric key with a key identifier which does not use the Receiver's FQDN, or if the Originator authenticated using a certificate, then the Receiver shall not include an *assignedSymmKeyID* attribute in the created resource.

4) The Receiver shall assign the *creator* attribute to an AE-ID or CSE-ID or Node-ID on instructions from the administrating stakeholder. The present document does not specify any details of how the AE-ID or CSE-ID or Node-ID is determined.

5) The Receiver shall instantiate the <*mefClientCmd*> child resource.

8.4.2.2 Retrieve

This procedure is denoted *MEF Client Configuration Retrieval* in clause 8.3.5.2.4 of oneM2M TS-0003 [2]. This procedure is used to retrieve the <*mefClientReg*> resource.

Originator:

No change from the generic procedures in clause 7.2.2.1 of [3] with clarifications discussed in clauses 5.2.2 and 6.3.

Receiver:

Same as the generic procedures in clause 7.2.2.2 of [3] with clarifications discussed in clauses 5.2.2 and 6.3, performing the following steps in order in the place of Recv-6.3: "Check authorization of the Originator":

1) The Receiver shall determine if the Originator is authorized by checking if the Originator is the creator of the resource:

   a) If the Originator is not authorized, then the Receiver shall execute the following steps in order:

      i) "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS_DENIED" error.

      ii) "Send the Response primitive".

   b) If the Originator is authorized, then the Receiver shall allow the request.

8.4.2.3 Update

This procedure is denoted *MEF Client Configuration Update* in clause 8.3.5.2.5 of oneM2M TS-0003 [2]. This procedure is used to update attributes of the <*mefClientReg*> resource, such as e.g. labels, expiration time.

Originator:

The <*mefClientReg*> resource shall not be updated by a MEF client via API.

Receiver:

Same as the generic procedures in clause 7.2.2.2 of [3] with clarifications discussed in clauses 5.2.2 and 6.3, and with the following differences:

The Receiver shall perform the following step in the place of Recv-6.3: "Check authorization of the Originator":

1) The Receiver shall determine if the Originator is authorized by checking if the Originator is the creator of the resource:

   a) If the Originator is not authorized, then the Receiver shall execute the following steps in order:

      i) "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS_DENIED" error.

      ii) "Send the Response primitive".

   b) If the Originator is authorized, then the Receiver shall allow the request.

The Receiver shall perform the following step as part of "Update the resource" (clause 7.3.3.7 of [3])" during Step Recv-6.5: "Create/Update/Retrieve/Delete/Notify operation is performed":

2) If the Originator was the Creator of the resource, then the Receiver shall perform steps 2 and 3 in clause 8.4.2.1.

8.4.2.4 Delete

This procedure is denoted *MEF Client De-Registration* in clause 8.3.5.2.6 of oneM2M TS-0003 [2]. This procedure enables the MEF client to delete its own <*mefClientReg*> resource on a MEF.

Originator:

No change from the generic procedures in clause 7.2.2.1 of [3] with clarifications discussed in clauses 5.2.2 and 6.3.

Receiver:

Same as the generic procedures in clause 7.2.2.2 of [3] with clarifications discussed in clauses 5.2.2 and 6.3, performing the following step in the place of Recv-6.3: "Check authorization of the Originator":

1) The Receiver shall determine if the Originator is authorized by checking if the Originator is the creator of the resource:

   a) If the Originator is not authorized, then the Receiver shall execute the following steps in order:

      i) "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS_DENIED" error.

      ii) "Send the Response primitive".

   b) If the Originator is authorized, then the Receiver shall allow the request.

8.5 Resource Type <*symmKeyReg*>

8.5.1 Introduction

A <*symmKeyReg*> resource shall represent a symmetric key registered with a MAF or MEF and administrated by the identified administrating stakeholder. A <*symmKeyReg*> resource shall be a child resource of a <*MAFBase*> or a <*MEFBase*> resource.

Table 8.5.1-1: Data Type Definition of <*symmKeyReg*>

Data Type ID	File Name	Note
symmKeyReg	SEC- symmKeyReg-v2_1_0.xsd	

Table 8.5.1-2: Universal/Common Attributes of <*symmKeyReg*> resource

Attribute Name	Request Optionality	
	Create	Update
@resourceName	NP	NP
resourceType	NP	NP
resourceID	NP	NP
parentID	NP	NP
creationTime	NP	NP
labels	O	O
creator	NP	NP
expirationTime	M	M

Table 8.5.1-3: Resource Specific Attributes of <*symmKeyReg*> resource

Attribute Name	Request Optionality		Data Type	Default Value and Constraints
	Create	Update		
adminFQDN	M	NP	xs:anyURI	No default
SUID	M	NP	m2m:suid	No default
targetIDs	O	O	m2m:listOfM2MID	No default
keyValue	O	NP	xs:hexBinary	No default

The <*symmKeyReg*> resource shall contain no child resources.

8.5.2 <*symmKeyReg*> resource specific procedures on CRUD operations

8.5.2.1 Create

This procedure is denoted *MAF Key Registration* in clause 8.8.2.7 of oneM2M TS-0003 [2] and *MEF Key Registration* in clause 8.3.5.2.7 of oneM2M TS-0003. This procedure enables a Source MAF Client or a Source MEF Client to establish a symmetric key with the MAF or MEF, respectively, which can be retrieved for use by one or more Target MAF Clients or Target MEF Clients.

Originator:

No change from the generic procedures in clause 7.2.2.1 of [3] with clarifications discussed in clauses 5.1.2 and 6.2 for Mmaf and in clauses 5.2.2 and 6.3 for Mmef, respectively, and with following differences:

In step Orig-1.0: "Compose of a Request primitive", the:

1) Originator shall select to either use a key derived from the TLS handshake or use another key provided by the Originator:

    a) If the Originator selects to use a key derived from the TLS handshake, then the Originator shall not include the *keyValue* attribute in the <*symmKeyReg*> resource of the request.

    b) If the Originator selects to provide a key other than a key derived from the TLS handshake, the Originator shall include the value of this key in the *keyValue* attribute in the <*symmKeyReg*> resource of the request.

In step Orig-6.0: "Process Response primitive", the following steps shall be performed:

2) If the Originator selected to use a key derived from the TLS handshake (see difference to step Orig-1.0 above), then the Originator shall perform the following steps in order to generate the value for the *keyValue* attribute:

    a) The Originator shall apply the TLS export mechanism described in clause 10.3.1 of [2] to generate a TLS-export-key. For MAF Key Registration the "TLS Key Export Details for M2M Secure Connection Key", for MEF Key Registration the "TLS Key Export Details for Enrolment Key" apply, respectively.

    b) The Originator shall apply the usage-constrained key derivation algorithm in clause 10.3.7 of [2] to derive the *keyValue* from TLS-export-key, *adminFQDN*, *SUID* and the *resourceName* assigned by the Receiver to the created resource.

3) The originator shall record the *resourceName* attribute of the created resource as the relative part of the key identifier for the symmetric key which is to be assigned to the value for the *keyValue* attribute.

Receiver:

Same as the generic procedures in clause 7.2.2.2 of [3] with clarifications discussed in clauses 5.2 and 6, and with following differences:

The Receiver shall perform the following steps in order in the place of Recv-6.3: "Check authorization of the Originator":

1) The Receiver shall ensure that the following criteria are satisfied, with administrating stakeholder being the stakeholder matching the *adminFQDN* attribute of the <*symmKeyReg*> resource in the Create request:

    a) The Originator is enrolled with the administrating stakeholder; that is, there is a non-expired <*mafClientReg*> resource whose *creator* attribute matches the Originator's AE-ID or CSE-ID or Node-ID, and whose *adminFQDN* attribute identifies the administrating stakeholder.

    b) The Receiver determines that the administrating stakeholder allows the creation of the resource. The present document does not specify how the Receiver makes this determination.

2) If these criteria are not met, then the Receiver shall execute the following steps in order:

    a) "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS_DENIED" error.

    b) "Send the Response primitive".

3) Otherwise, then the Receiver shall allow the request.

The Receiver shall perform the following steps in order as part of "Create the resource" (clause 7.3.3.5 of [3])" during Step Recv-6.5: "Create/Update/Retrieve/Delete/Notify operation is performed":

4) If the *keyValue* attribute is not present in the <*symmKeyReg*> resource in the request, then the Receiver shall perform the following step to generate the value for the *keyValue* attribute:

    a) The Receiver shall apply the TLS export mechanism described in clause 10.3.1 of [2] to generate a TLS-export-key.

b) The Receiver shall apply the usage-constrained key derivation algorithm in clause 10.3.7 of [2] to derive the value for the *keyValue* attribute from TLS-export-key, *adminFQDN*, *SUID* and the *resourceName* assigned by the Receiver to the created resource.

8.5.2.2 Retrieve

This procedure is denoted *MAF Key Retrieval* in clause 8.8.2.8 of oneM2M TS-0003 [2] and *MEF Key Retrieval* in clause 8.3.5.2.8 of oneM2M TS-0003. It enables a Target MAF Client to retrieve the Key Value from a MAF corresponding to a RelativeKeyID available to the Target MAF Client.

Originator:

No change from the generic procedures in clause 7.2.2.1 of [3] with clarifications discussed in clauses 5.1.2 and 6.2 for Mmaf and in clauses 5.2.2 and 6.3 for Mmef, respectively.

Receiver:

Same as the generic procedures in clause 7.2.2.2 of [3] with clarifications discussed in clauses 5.2 and 6, and with following differences:

The Receiver shall perform the following step in the place of Recv-6.3: "Check authorization of the Originator":

1) The Receiver shall determine if the Originator is authorized by checking if the Originator is the creator of the resource or the Originator is identified in the *targetIDs*:

   a) If the Originator is not authorized, then the Receiver shall execute the following steps in order:

      i) "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS_DENIED" error.

      ii) "Send the Response primitive".

   b) If the Originator is authorized, then the Receiver shall allow the request.

8.5.2.3 Update

This procedure is denoted *MAF Key Registration Update* in clause 8.8.2.9 of oneM2M TS-0003 [2] and *MEF Key Registration Update* in clause 8.3.5.2.9 of oneM2M TS-0003. It enables a Source MAF Client or Source MEF Client to update the metadata associated with a registered key.

Originator:

No change from the generic procedures in clause 7.2.2.1 of [3] with clarifications discussed in clauses 5.1.2 and 6.2 for Mmaf and in clauses 5.2.2 and 6.3 for Mmef, respectively.

Receiver:

Same as the generic procedures in clause 7.2.2.2 of [3] with clarifications discussed in clauses 5.2 and 6, and performing the following step in the place of Recv-6.3: "Check authorization of the Originator":

1) The Receiver shall determine if the Originator is authorized by checking if the Originator is the creator of the resource:

   a) If the Originator is not authorized, then the Receiver shall execute the following steps in order:

      i) "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS_DENIED" error.

      ii) "Send the Response primitive".

   b) If the Originator is authorized, then the Receiver shall allow the request.

8.5.2.4 Delete

This procedure is denoted *MAF Key De-Registration* in clause 8.8.2.10 of oneM2M TS-0003 [2] and *MEF Key De-Registration* in clause 8.3.5.2.10 of oneM2M TS-0003. It enables a Source MAF Client to request the MAF to stop distributing the registered key.

Originator:

No change from the generic procedures in clause 7.2.2.1 of [3] with clarifications discussed in clause 5.1.2 and 6.2 for Mmaf and in clauses 5.2.2 and 6.3 for Mmef, respectively.

Receiver:

Same as the generic procedures in clause 7.2.2.2 of [3] with clarifications discussed in clauses 5.2 and 6, and performing the following step in the place of Recv-6.3: "Check authorization of the Originator":

1) The Receiver shall determine if the Originator is authorized by checking if the Originator is the creator of the resource:

   a) If the Originator is not authorized, then the Receiver shall execute the following steps in order:

      i) "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS_DENIED" error.

      ii) "Send the Response primitive".

   b) If the Originator is authorized, then the Receiver shall allow the request.

8.6 Resource Type <*mefClientCmd*>

8.6.1 Introduction

A <*mefClientCmd*> resource shall represent a command to be indicated to a MEF client by a MEF, and a status report on the attempted parsing and execution of that command to be indicated to the MEF by a MEF Client. The retrieve procedure and update procedure are used for this purpose.

Table 8.6.1-1: Data Type Definition of <*mefClientCmd*>

Data Type ID	File Name	Note
mefClientCmd	SEC-mefClientCmd-v2_1_0.xsd	

Table 8.6.1-2: Universal/Common Attributes of <*mefClientCmd*> resource

Attribute Name	Request Optionality
	Update
@resourceName	NP
resourceType	NP
resourceID	NP
parentID	NP
creationTime	NP
labels	O
expirationTime	NP

Table 8.6.1-3: Resource Specific Attributes of *<mefClientCmd>* resource

Attribute Name	Request Optionality Update	Data Type	Default Value and Constraints
cmdID	M	m2m:requestID	No default
cmdDescription	NP	sec:cmdDescription	No default
cmdStatusCode	M	sec:cmdStatusCode	No default

The <*mefClientCmd*> resource shall contain no child resources.

8.6.2 <*mefClientCmd*> resource specific procedures on CRUD operations

8.6.2.1 Create

***Originator**:*

The <*mefClientCmd*> resource shall not be created via API. It is instantiated by a MEF when the parent <*mefClientReg*> resource is created as described in clause 8.4.2.1.

Receiver:

The primitive specific operation on Recv-1.0 "Check the syntax of received message" defined in TS-0004 [3] applies:

2) If the request is received, the Receiver CSE shall execute the following steps in order.

   a) "Create an unsuccessful Response primitive" with the ***Response Status Code*** indicating "OPERATION_NOT_ALLOWED" error.

   b) "Send the Response primitive".

8.6.2.2 Retrieve

This procedure is denoted *MEF Client Command Retrieval* in clause 8.3.9.2 of TS-0003 [2]. This procedure is used to retrieve the <*mefClientCmd*> resource.

Originator:

No change from the generic procedures in clause 7.2.2.1 of [3] with clarifications discussed in clauses 5.2.2 and 6.3, and with following difference:

In step Orig-6.0: "Process Response primitive", the Originator shall extract the *cmdID*, *cmdDescription* and *cmdStatusCode* from the response and pass these to the MEF Client Command processing as specified in clause 8.3.9.2 in oneM2M TS-0003 [2].

Receiver:

Same as the generic procedures in clause 7.2.2.2 of [3] with clarifications discussed in clauses 5.2.2 and 6.3, and with following differences:

The Receiver shall perform the following in the place of Recv-6.3: "Check authorization of the Originator":

1) The Receiver shall determine if the Originator is authorized by checking if the Originator is the creator of the parent <*mefClientReg*> resource.

   a) If the Originator is not authorized, then the Receiver shall execute the following steps in order.

      i) "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS_DENIED" error.

      ii) "Send the Response primitive".

b) If the Originator is authorized, then the Receiver shall allow the request.

The Receiver shall perform the following step as part of "Create the resource" (clause 7.3.3.5 of [2]) during Step Recv-6.5: "Create/Update/Retrieve/Delete/Notify operation is performed":

2) If the Receiver has a pending MEF Client Command to be issued to the Originator, then the Receiver shall set the *cmdID*, *cmdDescription* and *cmdStatusCode* attributes as specified in clause 8.3.9.2 of oneM2M TS-0003 [2]. The values of these attributes should remain set to these values until the MEF Client performs an Update on the Resource (see clause 8.6.2.3). If the MEF Client takes too long to perform an Update (for example, if the Response is not received by the Originator) then the MEF may choose to replace the attributes with a new MEF Client Command.

NOTE: The *cmdDescription* includes the *cmdClass* attribute which can be set to "NO_MORE_COMMANDS" by the MEF to indicate that there are no further commands to be issued.

8.6.2.3 Update

This procedure is denoted *MEF Client Command Update* in clause 8.3.9.3 of TS-0003 [2]. This procedure is used by the MEF Client to report on the status of an issued MEF Client Command, and for an MEF to issue another MEF Client Command.

Originator:

No change from the generic procedures in clause 7.2.2.1 of [3] with clarifications discussed in clauses 5.2.2 and 6.3, and with following differences:

In step Orig-1.0: "Compose of a Request primitive", the Originator shall include the *cmdID* and *cmdStatusCode* attributes in the Content of the request, with attribute values assigned as described in clause 8.3.9.3 of oneM2M TS-0003 [2].

In step Orig-6.0: "Process Response primitive", the Originator shall extract the *cmdID*, *cmdDescription* and *cmdStatusCode* attributes from the response and pass these to the MEF Client Command processing specified in clause 8.3.9.3 in oneM2M TS-0003 [2].

*Receiver***:**

Same as the generic procedures in clause 7.2.2.2 of [3] with clarifications discussed in clauses 5.2.2 and 6.3, and with the following differences:

The Receiver shall perform the following step in the place of Recv-6.3: "Check authorization of the Originator":

1. The Receiver shall determine if the Originator is authorized by checking if the Originator is the creator of the parent <*mefClientReg*> resource.

    a. If the Originator is not authorized, then the Receiver shall execute the following steps in order.

        i. "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS_DENIED" error.

        ii. "Send the Response primitive".

    b. If the Originator is authorized, then the Receiver shall allow the request.

The Receiver shall perform the following steps in order as part of "Update the resource" (clause 7.3.3.7) during Step Recv-6.5: "Create/Update/Retrieve/Delete/Notify operation is performed":

2. The Receiver shall extract the *cmdID* and *cmdStatusCode* attributes and pass these to the MEF Client Command processing in the MEF described in clause 8.3.9.3 of oneM2M TS-0003 [2].

3. If the Receiver has a pending MEF Client Command to be issued to the Originator, then the Receiver shall set the *cmdID*, *cmdDescription* and *cmdStatusCode* attributes as specified in clause 8.3.9.3 of oneM2M TS-0003 [2]. The values of these attributes should remain set to these values until the MEF Client performs a subsequent Update on the Resource. If the MEF Client takes too long to perform an Update (for example, if the

Response is not received by the Originator) then the MEF may choose to replace the attributes with a new MEF Client Command.

NOTE 2: The *cmdDescription* includes the *cmdClass* attribute which can be set to "NO_MORE_COMMANDS" to indicate that there are no further commands to be issued.

8.6.2.4 Delete

***Originator**:*

The <*mefClientCmd*> resource shall not be deleted via API. It is deleted by a MEF when the parent <*mefClientReg*> resource is deleted.

Receiver:

The primitive specific operation on Recv-1.0 "Check the syntax of received message" defined in TS-0004 [3] applies:

3) If the request is received, the Receiver CSE shall execute the following steps in order.

   a) "Create an unsuccessful Response primitive" with the ***Response Status Code*** indicating "OPERATION_NOT_ALLOWED" error.

   b) "Send the Response primitive".

9 Short Names

9.1 Introduction

The short names are introduced in clause 8.2.1 of oneM2M TS-0004 [3]. The short names in oneM2M TS-0004 [3] shall apply in addition to the short names defined here.

9.2 Security-specific oneM2M Resource attributes

In protocol bindings resource attributes names shall be translated into short names of table 9.2-1 and in table 8.2.3-1 of oneM2M TS-0004 [3].

Table 9.2-1: Security-specific oneM2M Attribute Short Names

Attribute Name	Occurs in	Short Name	Notes
resourceType	All	***ty****	Defined in oneM2M TS-0004 [3].
resourceID	All	***ri****	Defined in oneM2M TS-0004 [3].
resourceName	All	***rn****	Defined in oneM2M TS-0004 [3].
parentID	mafClientReg, mefClientReg, symmKeyReg	***pi****	Defined in oneM2M TS-0004 [3].
expirationTime	All	***et****	Defined in oneM2M TS-0004 [3].
creationTime	All	***ct****	Defined in oneM2M TS-0004 [3].
labels	mafClientReg, mefClientReg, symmKeyReg	***lbl****	Defined in oneM2M TS-0004 [3].
creator	mafClientReg, mefClientReg, symmKeyReg	***cr****	Defined in oneM2M TS-0004 [3].
adminFQDN	mafClientReg, mefClientReg, symmKeyReg	***adfq***	
SUID	symmKeyReg	***suid***	
assignedSymmKeyID	mafClientReg, mefClientReg	***aski***	
targetIDs	symmKeyReg	***tgis***	
keyValue	symmKeyReg	***kv***	
cmdID	mefClientCmd	***mcci***	
cmdDescription	mefClientCmd	***mccd***	
cmdStatusCode	mefClientCmd	***mccs***	
NOTE: Marked short names have been already assigned for primitive parameters or resource attributes in oneM2M TS-0004 [3].			

9.3 Security-specific oneM2M Resource types

In protocol bindings resource type names shall be translated into short names of table 9.3-1.

Table 9.3-1: Security-specific Resource Type Short Names

Attribute Name	Short Name
MAFBase	maf
MEFBase	mef
mafClientReg	macr
mefClientReg	mecr
symmKeyReg	mkr
mefClientCmd	mcc

9.4 Security-specific oneM2M Complex data type members

In protocol bindings complex data types member names shall be translated into short names of table 9.4-1.

NOTE: The member names of the security configuration parameters mefClientRegCfg, mafClientRegCfg, mefKeyRegCfg and mafKeyRegCfg are defined in clause 12.4 of oneM2M TS-0003 [3].

Table 9.4-1: Security-specific oneM2M Complex data type member short names

Member Name	Occurs in	Short Name	Notes
expirationTime	mefClientRegCfg, mefKeyRegCfg, mafClientRegCfg, mafKeyRegCfg	***et****	Defined in oneM2M TS-0004 [3]
labels	mefClientRegCfg, mefKeyRegCfg, mafClientRegCfg, mafKeyRegCfg	***lbl****	Defined in oneM2M TS-0004 [3]
fqdn	mefClientRegCfg, mefKeyRegCfg mafClientRegCfg, mafKeyRegCfg	***fq***	
adminFQDN	mefClientRegCfg, mafClientRegCfg	***adfq****	
httpPort	mefClientRegCfg, mafClientRegCfg	***hpt***	
coapPort	mefClientRegCfg, mafClientRegCfg	***cpt***	
websocketPort	mefClientRegCfg, mafClientRegCfg	***wpt***	
ppsk	mefClientRegCfg, mafClientRegCfg	***pk***	
rpsk	mefClientRegCfg, mafClientRegCfg	***rk***	
certAuth	mefClientRegCfg, mafClientRegCfg	***cert***	
credID	mefClientRegCfg, mafClientRegCfg	***crdi***	
caCerts	mefClientRegCfg, mafClientRegCfg	***cact***	
SUID	mefKeyRegCfg, mafClientRegCfg, authProfileMONodeArgs	***suid****	
targetIDs	mefKeyRegCfg, mafClientRegCfg	***tgis***	
targetID	cmdDescription	***tgi***	
cmdClassID	cmdDescription	***ccid***	
cmdArgs	cmdDescription	***cma***	
certProvProtocolID	certProvCmdArgs	***cppi***	
URI	certProvCmdArgs	***uri****	
certSubjectType	certProvCmdArgs	***cst***	
certSubjectID	certProvCmdArgs	***csi***	
deviceConfigURI	devCfgCmdArgs	***dcu***	
objectPath	MONodeCmdArgs	***ajop****	
objectTypeID	MONodeCmdArgs	***otyp***	
objectTypeSpecificArgs	MONodeCmdArgs	***otsa***	
retryDuration	noMoreCmdArgs	***rdu***	
noMoreCmdArgs	cmdArgs	***nmca***	
certProvCmdArgs	cmdArgs	***cpca***	
devCfgCmdArgs	cmdArgs	***dcca***	
MONodeCmdArgs	cmdArgs	***nnca***	
NOTE: * marked short names have been already assigned to an attribute in table 9.2-1.			

History

Publication history		
V3.0.0	February 2019	Release 3 - Publication