|  |  |
| --- | --- |
| Input Contribution | |
| Meeting ID\* | SEC#27 |
| Title:\* | Update to TS-0032v0\_0\_2, MAF and MEF Interface Specification |
| Source:\* | Wolfgang Granzow, Qualcomm, [wgranzow@qti.qualcomm.com](mailto:wgranzow@qti.qualcomm.com)  Phil Hawkes, Qualcomm, [phawkes@qti.qualcomm.com](mailto:phawkes@qti.qualcomm.com) |
| Uploaded Date:\* | 2017-02-05 |
| Document(s)  Impacted\* | TS-0032v0\_0\_2 |
| Intended purpose of  document:\* | Decision  Discussion  Information  Other <specify> |
| Decision requested or recommendation:\* | Update text in TS-0032v0\_0\_2 as proposed in this contribution. |
| Template Version:23 February 2015 (Do not modify) | |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

**Introduction**

This contribution provides following updates to TS-0032v0\_0\_2:

1. Connection and alignment of the resource type specific procedures with text already existing in TS-0003
2. Addition of clause 9 with short names (based on earlier contribution SEC-2016-0138)
3. Minor editorial corrections and additions

*======== Text proposed for TS-0032 starts here =============================*

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”.

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

3.2 Abbreviations

MAF M2M Authentication Function

MEF M2M Enrolment Function

MTE M2M Trust Enabler

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 [i.1]

5 General Description

*Editor’s Note: The subclauses below will be updated when the MEF interface will be introduced*

5.1 Introduction to the MAF Interface

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 Node (ADN, ASN, IN or MN), or a CSE or an AE.

Figure 5.1-1 defines the reference point Mmaf between MAF clients and a MAF.



**Figure 5.1-1: 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 specification has no impact on the specifications in TS-0001 [1] and TS-0004 [3]. However, the MAF Interface uses much of the specification in TS-0004 [3] and in particular allows use of the HTTP binding in TS-0008 [4], the CoAP binding in TS-0009 [5] and the WebSocket binding in TS-0020 [8].

NOTE: The MQTT binding in 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.

5.2 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].

Table 5.2-1 and 5.2-2 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.

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 5.2-1: 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 |  |
| Content | 0..1 |  |
| Result Content | 0..1 |  |

**Table 5.2-2: 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 5.2-3 lists the response status codes from clause 6.6 [3] which are supported by the MAF Interface.

**Table 5.2-3: 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 |

*Editor’s note: Unclear which, if any, of the oneM2M specific MIME media types from clause 6.7 [3] are supported.*

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

6 Processing and Representation of Primitives

*Editor’s Note: This clause will be updated when the MEF interface will be introduced*

This clause is based on the specification in clause 7 and 8 of oneM2M TS-0004 [3].

The MAF Interface request primitive format conforms to clause 7.2.1.1 [3], constrained to the CRUD operations, with request parameters listed in Table 5.2-1.

The MAF Interface response primitive format conforms to clause 7.2.1.2 [3], constrained to the CRUD operations, with response parameters listed in Table 5.2-2.

The MAF Interface generic resource request procedure for originators and receivers conforms to clauses 7.2.2.1 and 7.2.2.2 of [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 [3]) 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 5.2-1 and Table 5.2-2.
* “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 [3], with clause 7.3.3.15 [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 [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 [3] does not apply to the MAF Interface.

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

The representation of MAF Interface primitives in data transfer conforms to Clause 8. Clause 9 (tbc.) contains additional short names specific to the MAF Interface.

*Editor’s note: the clause that defines the short names will be added*

7 Resource types definitions

7.1 Resource Type <*MAFBase*>

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

The *<MAFBase>* resource shall contain the child resources specified in table 7.1-1.

**Table 7.1-1: Child resources of *<MAFBase>* resource**

| **Child Resources of *<MAFBase>*** | **Child Resource Type** | **Multiplicity** | **Description** |
| --- | --- | --- | --- |
| *[variable]* | *<mafClientReg>* | 0..n | See clause 7.2 |
| *[variable]* | *<symmKeyReg>* | 0..n | See clause 7.3 |

The *<MAFBase>* resource shall contain the attributes specified in table 7.1-2.

**Table 7.1-2: 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”. |
| *expirationTime* | 1 | RO | See clause 9.6.1.3 of [1]. |
| *creationTime* | 1 | RO | See clause 9.6.1.3 of [1]. |

7.2 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 may 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 MAF Client may be associated with multiple *<mafClientReg>* resources on multiple MAFs.

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

The *<mafClientReg>* resource shall contain the attributes specified in table 7.2-1.

**Table 7.2-1: 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]. |
| *expirationTime* | 1 | WO | 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]. |
| *creator* | 1 | WO | See clause 9.6.1.3 of [1]. *Editor's note: a new attribute may need to be defined which allows Node-ID in addition to CSE-ID and AE-ID* |
| *fqdn* | 1 | WO | FQDN of the M2M SP or MTE who is the administrating stakeholder of this enrolment |
| *assignedSymmKeylID* | 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. |

7.3 Resource Type <*symmKeyReg*>

The <*symmKeyReg*> resource shall represent a symmetric key that a source MAF Client has established with the MAF for distributing to authorized Target MAF Clients and/or another MAF. The MAF Client provides a list of authorized Targets when the resource is created – the present document does not specify how the MAF associates the list with the resource. The MAF, is 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 7.3-1.

**Table 7.3-1: 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]. |
| *expirationTime* | 1 | WO | 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]. |
| *creator* | 1 | RO | See clause 9.6.1.3 of [1]. |
| *fqdn* | 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 fqdn) 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 Client and MAF from the TLS handshake parameters. |

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.

The <*MAFBase*> resource has no attributes.

**Table 8.1.1-1: Child resources of <MAFBase> resource**

|  |  |  |  |
| --- | --- | --- | --- |
| **Child Resource Type** | **Child Resource Name** | **Multiplicity** | **Ref. to Resource Type Definition** |
| *<mafClientReg>* | [variable] | 0..n | Clause 7.2 |
| <*symmKeyReg*> | [variable] | 0..n | Clause 7.3 |

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":

If the request is received, the MAF shall execute the following steps in order.

1. "Create an unsuccessful Response primitive" with the ***Response Status Code*** indicating "OPERATION\_NOT\_ALLOWED" error.
2. "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 5.2 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”:

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.
2. "Create an unsuccessful Response primitive" with the Response Status Code indicating "OPERATION\_NOT\_ALLOWED" error.
3. "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.
2. "Create an unsuccessful Response primitive" with the ***Response Status Code*** indicating "OPERATION\_NOT\_ALLOWED" error.
3. "Send the Response primitive".

8.2 Resource Type *<mafClientReg>*

8.2.1 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 8.2.1-1: 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 |
| *expirationTime* | M | M |
| *labels* | O | O |
| *creator* | NP | NP |

**Table 8.2.1-2: Resource Specific Attributes of *<mafClientReg>* resource**

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

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

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

8.2.2.1 Create

This procedure is denoted *MAF Client Registration* in clause 8.8.2.3 of 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.

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 (cf. clause 5.1), the Node-ID of the respective ADN, ASN, MN or IN shall serve as MAF Client ID.

*Editor’s Note: the applicable format(s) of the MAF Client ID require more clarification*

***Originator****:*

No change from the generic procedures in clause 7.2.2.1 of [3] with clarifications discussed in clauses 5.2 and 6, 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* then the originator shall use the *assignedSymmKeyID* to identify this symmetric key when it is subsequently used in authenticating to the MAF.

*Editor's note: May need specific text to allow for Node-ID in* ***From****.*

***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 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.
   1. If the Originator is not authorized, then the Receiver shall execute the following steps in order.
      1. "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS\_DENIED" error.
      2. "Send the Response primitive".
   2. 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)” during Step Recv-6.5: “Create/Update/Retrieve/Delete/Notify operation is performed”:

1. If the Originator authenticated using symmetric key with a key identifier which does not use the Receiver’s FQDN, then
   1. 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.
   2. 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.
2. 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.
3. 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.
4. The Receiver may assign the *mafClientCfg* attribute on instructions from the administrating stakeholder.

8.2.2.2 Retrieve

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

*Editor’s Note: Should “partial retrieve” of resources (retrieve of individual attributes) be supported by the MAF interface (see clause 7.3.3.6 of TS-0004 [3]) or shall we impose a restriction that the MAF interface supports “full” retrieve only?*

*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 6.3, 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.

*Editor's note: May need specific text to allow for Node-ID in* ***From*** *and the creator attribute.*

1. If the Originator is not authorized, then the Receiver shall execute the following steps in order.
   * 1. "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS\_DENIED" error.
     2. "Send the Response primitive".
2. If the Originator is authorized, then the Receiver shall allow the request.

8.2.2.3 Update

This procedure is denoted *MAF Client Configuration Update* in clause 8.8.2.5 of 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 5.2 and 6, 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.

*Editor's note: May need specific text to allow for Node-ID in* ***From*** *and the creator attribute.*

1. If the Originator is not authorized, then the Receiver shall execute the following steps in order.
   * 1. "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS\_DENIED" error.
     2. "Send the Response primitive".
2. 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”:

1. If the Originator was the Creator of the resource, then the Receiver shall perform steps 2 and 3 in clause 8.2.2.1.

8.2.2.4 Delete

This procedure is denoted *MAF Client De-Registration* in clause 8.8.2.6 of 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 5.2 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, 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.

*Editor's note: May need specific text to allow for Node-ID in* ***From*** *and the creator attribute.*

1. If the Originator is not authorized, then the Receiver shall execute the following steps in order.
   * 1. "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS\_DENIED" error.
     2. "Send the Response primitive".
2. If the Originator is authorized, then the Receiver shall allow the request.

8.3 Resource Type <*symmKeyReg*>

8.3.1 Introduction

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

**Table 8.3.1-1: 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 |
| *expirationTime* | M | M |
| *creator* | NP | NP |
| *labels* | O | O |

**Table 8.3.1-2: Resource Specific Attributes of *<symmKeyReg>* resource**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Request Optionality** | | **Data Type** | **Default Value and Constraints** |
| **Create** | **Update** |
| *fqdn* | M | NP | xs:anyURI | No default |
| *SUID* | M | NP | m2m:suid | No default |
| *targetIDs* | O | O | m2m:listOfM2MID | No default |
| *keyValue* | O | NP | xs:base64binary | No default |

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

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

8.3.2.1 Create

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

*Originator:*

No change from the generic procedures in clause 7.2.2.1 of [3] with clarifications discussed in clauses 5.2 and 6, 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.
   1. 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.
   2. 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

1. 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
   1. The Originator shall apply the TLS export mechanism described in clause 10.3.1 of [2] to generate a TLS-export-
   2. 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, *fqdn, SUID* and the *resourceName* assigned by the Receiver to the created resource*.*
2. 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 *fqdn* attribute of the <*symmKeyReg*> resource in the Create request:
   1. 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 *fqdn* attribute identifies the administrating stakeholder.

*Editor's note: May need specific text to allow for Node-ID in* ***From*** *and the creator attribute.*

* 1. 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.

1. If these criteria are not met, then the Receiver shall execute the following steps in order.
   1. "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS\_DENIED" error.
   2. "Send the Response primitive".
2. 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)” during Step Recv-6.5: “Create/Update/Retrieve/Delete/Notify operation is performed”:

1. If the *keyValue* attribute is not present in the <*symmKeyReg*> resource in the request, then the Receiver shall perform the following steps in order to generate the value for the *keyValue* attribute
   1. The Receiver shall apply the TLS export mechanism described in clause 10.3.1 of [2]to generate a TLS-export-key.
   2. 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, *fqdn, SUID* and the *resourceName* assigned by the Receiver to the created resource*.*

8.3.2.2 Retrieve

This procedure is denoted *MAF Key Retrieval* in clause 8.8.2.8 of TS-0003 [2]. 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 5.2 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 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 by checking if the Originator is the creator of the resource or the Originator is identified in the *targetIDs*.

*Editor's note: May need specific text to allow for Node-ID in* ***From,*** *and the creator attribute.*

1. If the Originator is not authorized, then the Receiver shall execute the following steps in order.
   * 1. "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS\_DENIED" error.
     2. "Send the Response primitive".
2. If the Originator is authorized, then the Receiver shall allow the request.

8.3.2.3 Update

This procedure is denoted *MAF Key Registration Update* in clause 8.8.2.9 of TS-0003 [2]. It enables a Source MAF 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 5.2 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 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.

*Editor's note: May need specific text to allow for Node-ID in* ***From,*** *and the creator attribute.*

1. If the Originator is not authorized, then the Receiver shall execute the following steps in order.
   * 1. "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS\_DENIED" error.
     2. "Send the Response primitive".
2. If the Originator is authorized, then the Receiver shall allow the request.

8.3.2.4 Delete

This procedure is denoted *MAF Key De-Registration* in clause 8.8.2.10 of TS-0003 [2]. 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 5.2 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 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.

*Editor's note: May need specific text to allow for Node-ID in* ***From,*** *and the creator attribute.*

1. If the Originator is not authorized, then the Receiver shall execute the following steps in order.
   * 1. "Create an unsuccessful Response primitive" with the Response Status Code indicating "ACCESS\_DENIED" error.
     2. "Send the Response primitive".
2. If the Originator is authorized, then the Receiver shall allow the request.

## 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, 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, symmKeyReg | ***lbl*** | Defined in oneM2M TS-0004 [3]. |
| *creator* | mafClientReg, symmKeyReg | ***cr*** | Defined in oneM2M TS-0004 [3]. |
| *fqdn* | mafClientReg, symmKeyReg | ***fq*** |  |
| *SUID* | symmKeyReg | ***suid*** |  |
| *assignedSymmKeylID* | mafClientReg | ***aski*** |  |
| *mafClientCfg* | mafClientReg | ***ccfg*** |  |
| *targetIDs* | symmKeyReg | ***tgis*** |  |
| *keyValue* | symmKeyReg | ***kv*** |  |
| NOTE: Marked short names have been already assigned for primitive parameters 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 |
| *mafClientReg* | mcr |
| *symmKeyReg* | mkr |

## 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 tefclientCfg, tefClientRegCfg and tefKeyRegCfg are defined in clause 12.4 of TS-0003 [3].

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

| Member Name | Occurs in | Short Name | Notes |
| --- | --- | --- | --- |
| tefClientRegCfg | tefClientCfg | ***tcrc*** |  |
| tefKeyRegCfg | tefClientCfg | ***tkrc*** |  |
| expirationTime | tefClientRegCfg, tefKeyRegCfg | ***et\**** | Defined in oneM2M TS-0004 |
| labels | tefClientRegCfg, tefKeyRegCfg | ***lbl\**** | Defined in oneM2M TS-0004 |
| fqdn | tefClientRegCfg, tefKeyRegCfg | ***fq\**** |  |
| URI | tefClientRegCfg | ***uri*** | Defined in oneM2M TS-0004 [3] |
| httpPort | tefClientRegCfg | ***hpt*** |  |
| coapPort | tefClientRegCfg | ***cpt*** |  |
| websocketPort | tefClientRegCfg | ***wpt*** |  |
| ppsk | tefClientRegCfg | ***pk*** |  |
| rpsk | tefClientRegCfg | ***rk*** |  |
| certAuth | tefClientRegCfg | ***cert*** |  |
| credID | tefClientRegCfg | ***crdi*** |  |
| caCerts | tefClientRegCfg | ***cact*** |  |
| SUID | tefKeyRegCfg | ***suid\**** |  |
| targetIDs | tefKeyRegCfg | ***tgis*** |  |
| NOTE: \* marked short names have been already assigned to an attribute in Table 9.2-1. | | | |