|  |
| --- |
| i |

|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | SDS #49 |
| Source:\* | Andreas Kraft, DT, A.Kraft@telekom.de  Andreas Neubacher, DT, [Andreas.Neubacher@magenta.at](mailto:Andreas.Neubacher@magenta.at) |
| Date:\* | 2021-02-01 |
| Reason for Change/s:\* | Editorial corrections for TS-0001 (R3) |
| CR against: Release\* | Release 3 |
| CR against: WI\* | Active WI-xxxx  MNT maintenance / < Work Item number(optional)>  Is this a mirror CR? Yes  No  mirror CR number: (Note to Rapporteur - use latest agreed revision)  STE Small Technical Enhancements / < Work Item number (optional)>  Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0001, V3.22 |
| Clauses \* | 9.6.2.2, 9.6.2.4, 11.1, 11.3.4, 11.5.1 |
| Type of change: \* | Editorial change  Bug Fix or Correction  Change to existing feature or functionality  New feature or functionality  Only ONE of the above shall be ticked |
| Impacted other TS/TR(s) |  |
| Post Freeze checking:\* | This CR contains only essential changes and corrections? YES  NO  This CR may break backwards compatibility with the last approved version of the TS? YES  NO |
| Template Version: January 2017 (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.

GUIDELINES for Change Requests:

Provide an informative introduction containing the problem(s) being solved, and a summary list of proposals.

Each CR should contain changes related to only one particular issue/problem.

In case of a correction, and the change apply to previous releases, a separate “mirror CR” should be posted at the same time of this CR

Mirror CR: applies only when the text, including clause numbering are exactly the same.

Companion CR: applies when the change means the same but the baselines differ in some way (e.g. clause number).

Follow the principle of completeness, where all changes related to the issue or problem within a deliverable are simultaneously proposed to be made E.g. A change impacting 5 tables should not only include a proposal to change only 3 tables. Includes any changes to references, definitions, and acronyms in the same deliverable.

Follow the drafting rules.

All pictures must be editable.

Check spelling and grammar to the extent practicable.

Use Change bars for modifications.

The change should include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change. Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the new clause is proposed to be located.

Multiple changes in a single CR shall be clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.

When subsequent changes are made to content of a CR, then the accepted version should not show changes over changes. The accepted version of the CR should only show changes relative to the baseline approved text.

Introduction

This CR proposes a couple of editorial corrections for TS-0001.

Change 1: in “Table 9.6.2.4-1: Types of Parameters in accessControlObjectDetails” the attribute “specialization” should be named “specializationType”. This would then be consistent with, for example, the definition in TS-0004, “Table 6.3.5.27 1: Type Definition of m2m:accessControlRule”. This is corrected in this change.

Change 2: in “Table 9.6.2.2-1: Types of Parameters in accessControlContexts” the attribute “accessControlIpAddress” should be named “accessControlIpAddresses”. This would then be consistent with, for example, the definition in TS-0004, “Table 6.3.5.27 1: Type Definition of m2m:accessControlRule”. This is corrected in this change.

Change 3,4,5 : The attribute “owner” was renamed “holder”, but the attribute is still named “owner” in various places. This is corrected in these changes.

This is a mirror for SDS-2021-0001 for R3.

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of Change 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### 9.6.2.4 accessControlObjectDetails

The *accessControlObjectDetails* is an optional parameter of an access control rule. It specifies a subset of child resource types of the targeted resource to which the access control rule applies. If an access control rule includes *accessControlObjectDetails*, then *childResourceType* shall be specified. An access control rule which does not include any *accessControlObjectDetails* parameters applies to the child resource types of the target resource. The *accessControlObjectDetails* parameter shall consist of the elements listed in table 9.6.2.4-1. Child resource types listed in the *childResourceType* component are subject of access control for the Create operation only. Once a child resource is created, the Access Control Policies assigned directly to it apply. The *resourceType* and *specializationType* element are optional. If either the *resourceType* or *specializationType* element is present in *accessControlObjectDetails*, the CSE shall match the type of resource or specialization of the targeted resource with the value specified in the *resourceType* or *specializationType* element. Further checking of *childResourceType* shall be done only if the *resourceType* or *specializationType* match occurs. However, if the *resourceType* and *specializationType* elements are not provided, only *childResourceType* match shall be performed.

Table 9.6.2.4-1: Types of Parameters in *accessControlObjectDetails*

| **Name** | **Description** |
| --- | --- |
| *resourceType* | Identifier of the resource type to which this access control rule applies |
| *specializationType* | When the *resourceType* is *mgmtObj* or *flexContainer*, the identifier of the specialization as defined by *mgmtDefinition* or *containerDefinition* attribute, respectively, shall be specified. |
| *childResourceType* | List of child resource types and/or the identifier of the specialization. The identifier of the specialization shall be specified when the *resourceType* is *mgmtObj* or *flexContainer*. |

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of Change 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of Change 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### 9.6.2.2 *accessControlContexts*

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

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

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

| Name | Description |
| --- | --- |
| *accessControlTimeWindow* | Represents a time window constraint which is compared against the time that the request is received at the Hosting CSE. |
| *accessControlLocationRegion* | Represents a location region constraint which is compared against the location of the Originator of the request. |
| *accessControlIpIPAddresses* | Represents an IP address constraint or IP address block constraint which is compared against the IP address of the Originator of the request. |

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of Change 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of Change 3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 11.1 Enrolling M2M Nodes and M2M Applications for oneM2M Services

Though M2M Nodes in the field domain are assumed to communicate without human involvement, individuals or organizations remain responsible for setting the access control policies used to authorize their M2M Nodes to access M2M services. In the following text, M2M Nodes refers to M2M field nodes.

In particular, individuals or organizations acquiring M2M Nodes can subscribe to a contract with an M2M Service provider (M2M Service Subscription) under which they enrol their M2M Nodes (e.g. using identifiers pre-provisioned on the nodes, such as Node-ID). This in turn may require an M2M Service provisioning step (including Security provisioning) that takes place on the target M2M Nodes themselves, for which interoperable procedures are specified by oneM2M (see clause 11.2.1). Following M2M service provisioning, the nodes can be identified and authenticated for association with an M2M Service Subscription, whose properties reflect the contractual agreement established between their holder and the M2M Service Provider.

Similarly, it may be possible for an M2M Service Provider to mandate that an M2M Application accessing M2M services be associated with a security credentials used to authorize specific operations to instance of that M2M Application, i.e. AEs (see clause 11.2.2). This step facilitates the deployment and management of M2M Applications that are instantiated in great numbers, as it enables all instances of an M2M Application to be managed through common security policies that are set once for all. It also enables keeping control over M2M Applications issued by untrusted sources.

The above steps may be delegated to an M2M trust enabler, when this role is not assumed by the M2M Service Provider.

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of Change 3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of Change 4 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 11.3.4 M2M Authorization Procedure

The M2M authorization procedure controls access to resources and services by CSEs and AEs. This procedure requires that the Originator has been identified to an M2M Authentication Function and mutually authenticated and associated with an M2M Service Subscription. Authorization depends on:

* The privileges set by the M2M Service Subscription associated with the Originator (e.g. service/role assigned to the Originator).
* These privileges are set-up based on the access control policies associated with the accessed resource or service. They condition the allowed operations (e.g. CREATE) based on the Originator's privileges and other access control attributes (e.g. contextual attributes such as time or geographic location).
* Role-IDs which have been associated with the Originator.

The authorization/access grant involves an Access Decision step to determine what the authenticated CSE or AE can actually access, by evaluating applicable access control policies based on the CSE or AE privileges. Access Decision is described in oneM2M TS-0003 [2].

The following set of access control policy attributes shall be available for an Access Decision.

* Access control attributes of Originator and Originator's Role (e.g. Role-IDs, CSE\_IDs, AE-IDs, etc.).
* Access control attributes of Environment/Context (e.g. time, day, IP address, etc.).
* Access control attributes of Operations (e.g. Create, Execute, etc.).

The M2M Service Provider/administrator and holder of resources are responsible to establish access control policies that determine by whom, in what context and what operations may be performed upon those resources. If the request satisfies the holder’s access control policy, then the access to the resource is granted.

**Dynamic Authorization:** Dynamic Authorization encompasses:

1. authorizing the creation of a limited-lifetime access control policy authorizing the Originator to perform specific operations on the requested resource; and
2. issuing limited-lifetime Tokens associating the Originator with Role-IDs and/or access control policies for identified resources.

Two forms of Dynamic Authorization are supported: Direct Dynamic Authorization and Indirect Dynamic Authorization.

In the event that the request does not satisfy any of the holder’s access control policies, then Dynamic Authorization may be requested from Dynamic Authorization System (DAS) Servers; this is called *Direct Dynamic Authorization*,and relevant details are provided clause 11.5.2. The request is then re-evaluated to determine if the holder’s access control policy is now satisfied and access is granted.

If access is still denied, then the Originator is provided with ***Token Request Information*** used to request the issuance of Tokens by a Dynamic Authorization System. A Token identifies Role-IDs and/or access control policies (for identified resources) which have been temporarily associated with the Originator. The Originator then resends the request from the Originator, this time adding any Token or Token-IDs received from the Dynamic Authorization System. This is called *Indirect Dynamic Authorization*, andrelevant details are provided clause 11.5.3.

NOTE: A DAS Server can be triggered, by Dynamic Authorization, to update the access control policy configuration using oneM2M request primitives.

In the event that the requesting entity does not satisfy the holder’s access control policy, a Hosting CSE shall check to see if the resource (or one of its parents) has a *dynamicAuthorizationConsultationIDs* which links to a valid *<dynamicAuthorizationConsultation>* resource. If there is no valid *<dynamicAuthorizationConsultation>* resource or if the dynamicAuthorizationEnabled attribute is set to "false", then then the Hosting CSE shall not attempt to perform direct dynamic authorization on behalf of the requesting entity. However, if there is a valid *<dynamicAuthorizationConsultation>* resource available and if the dynamicAuthorizationEnabled attribute is set to "true", then the Hosting CSE shall initiate a direct dynamic authorization request to the specified dynamicAuthorizationPoA. If direct dynamic authorization results in sufficient privileges being granted to the requesting entity, the Hosting CSE shall grant it access. In addition the Hosting CSE may also dynamically create a new access control policy and configure it with the granted privileges along with any specified lifetime associated with the privileges based on a resource creation process initiated by the dynamic authorization system.

This function shall fetch the subscription related information in order to check if a Role-ID used in a request is allowed by the M2M service subscription. The authorization procedure shall be implemented as specified in the oneM2M TS‑0003 [2].

**Distributed Authorization**

A distributed authorization system may comprise four functional components: Policy Enforcement Point (PEP), Policy Decision Point (PDP), Policy Retrieval Point (PRP) and Policy Information Point (PIP). A PEP that coexists with the Hosting CSE enforces the access control decision. PDP, PRP and PIP are responsible for making access control decisions, providing applicable access control policies and obtaining access control information required by access control policy evaluation procedures respectively. In a distributed authorization system these components may be distributed in different CSEs. Details of these components are described in oneM2M TS-0003 [2].

Three resource types are defined for representing PDPs, PRPs and PIPs: <*authorizationDecision*>, <*authorizationPolicy*> and <*authorizationInformation*>. For details about these resource types see clause 9.6.41, 9.6.42 and 9.6.43.

Three attributes are defined in the *<accessControlPolicy>* resource type for providing the addresses of PDPs, PRPs and PIPs: *authorizationDecisionResourceIDs*, *authorizationPolicyResourceIDs* and *authorizationInformationResourceIDs*. For details about these resource attributes see clause 9.6.2.

A high level description of the distributed authorization framework and procedures is provided in clause 11.6.

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of Change 4 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of Change 5 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 11.5.1 Dynamic Authorization Reference Model

The Dynamic Authorization reference model is shown in figure 11.5.1-1



Figure 11.5.1-1: Dynamic Authorization reference model

The Dynamic Authorization reference model introduces the following systems and entities:

* Dynamic Authorization System (DAS): A system supporting dynamically authorization on behalf of resources holders. The present document does not describe the processing and exchange of messages within the Dynamic Authorization System. This system may reside either internally or externally within the service provider network.
* Dynamic Authorization System (DAS) Server: A server configured with policies for dynamic authorization, and provided with credentials for issuing Tokens. The DAS Server may include an AE for interaction with the oneM2M system.

The following Dynamic Authorization procedures are specified:

* **Direct Dynamic Authorization**, summarized in figure 11.5.1-2. In this procedure, Hosting CSE interacts with the DAS Server to obtain Dynamic Authorization. When AE, Hosting CSE and the DAS server support to create the Authorization Relationship Mapping Record, steps 5-7 will be applied.



Figure 11.5.1-2: Direct Dynamic Authorization

* **Indirect Dynamic Authorization**, summarized in figure 11.5.1-3:
* Steps 1-2: The Hosting CSE may provide the Originator with ***Token Request Information*** in the unsuccessful response.
* Steps 3: The Originator interacts with the DAS Server with the intention that the DAS Server issue *Tokens* authorizing the Originator, and the Originator is provided with the Token or a Token-ID. If the Originator is an AE, whose AE-ID-Stem is assigned by the registrar CSE, and both AE and DAS server support to create the Authorization Relationship Mapping Record, the DAS Server shall request the AE to create the authorization relationship mapping record. The interaction is not described in the present specification.
* Steps 4: If the DAS Server starts the process of Authorization Relationship Mapping Record creation in step 3, the AE shall request to create the Authorization Relationship Mapping Record in the DAS Server.
* Steps 5-8: The Originator provides the Hosting CSE with a *Token, Token-ID* to indicate that the Token is to be considered in the access decision. In the case of a token-ID, the Hosting CSE retrieves the corresponding Token via an AE of the DAS Server. These are then used in the access decision. If the Authorization Relationship Mapping Record is created in step 4, the originator shall also indicate the related information to the Hosting CSE. The Hosting CSE may provide the Originator with a *Local-Token-ID* may be used to identify the Token.



Figure 11.5.1-3: Indirect Dynamic Authorization

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of Change 5 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*