|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting:\* | ARC#26.3 |
| Source:\* | C-DOT |
| Date:\* | 2017-01-09 |
| Contact:\* | Poornima (poornima@cdot.in), Chaitan(chaitan.yadav@cdot.in),Suman(ssheoran@cdot.in )  |
| Reason for Change/s:\* | See the introduction  |
| CR against: Release\* | Release 3 |
| CR against: WI\* | [ ]  Active <Work Item number> [x]  MNT maintenace / < Work Item number(optional)>[ ]  STE Small Technical Enhancements / < Work Item number (optional)>Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0001 v3.2.0 |
| Clauses/Sub Clauses\* | Section 9.6.2 |
| Type of change: \* | [ ]  Editorial change[x]  Bug Fix or Correction[ ]  Change to existing feature or functionality[ ]  New feature or functionalityOnly ONE of the above shall be ticked |
| Post Freeze checking:\* | This CR contains only essential changes and corrections? YES [x]  NO [ ] This CR may break backwards compatibility with the last approved version of the TS? YES [x]  NO [ ] This CR is a mirror CR? YES [x]  if YES, please indicate the document number of the original CR:<ARC-2016-0472R01> : NO [ ]   |
| Template Version:27 May 2015 (Dot 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 separated “mirror CR” should be posted at the same time of this CR

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 sections need not show surrounding clauses as long as the proposed section number clearly shows where the new section 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

### During the discussion of CR ARC-2016-0462 (TP 25) , it was discussed that for updation of acpId attribute of any resource , permission in selfPrivileges is being checked by Hosting CSE, while for updation of other attributes of a resource, permission in Privileges is being checked .

In TS-0001, section 9.6.1.3.2 , accessControlPolicyIDs description, it is mentioned as highlighted below

|  |  |
| --- | --- |
| *accessControlPolicyIDs* | The attribute contains a list of identifiers of an *<accessControlPolicy>* resource. The privileges defined in the *<accessControlPolicy>* resource that are referenced determine who is allowed to access the resource containing this attribute for a specific purpose (e.g. Retrieve, Update, Delete, etc.).If a resource type does not have an *accessControlPolicyIDs* attribute definition, then the *accessControlPolicyIDs* for that resource is governed in a different way, for example, the *accessControlPolicy* associated with the parent may apply to a child resource that does not have an *accessControlPolicyIDs* attribute definition, or the privileges for access are fixed by the system. Refer to the corresponding resource type definitions and procedures to see how access control is handled in such cases.If a resource type does have an *accessControlPolicyIDs* attribute definition, but the (optional) *accessControlPolicyIDs* attribute is not set, or it is set to a value that does not correspond to a valid, existing *<accessControlPolicy>* resource, or it refers to an *<accessControlPolicy>* resource that is not reachable (e.g. because it is located on a remote CSE that is offline or not reachable), then the system default access privileges shall apply.All resources are accessible if and only if the privileges (i.e. shored as *privileges* or *selfPrivileges* attribute of <accessControlPolicy> resource) allow it, therefore all resources shall have an associated *accessControlPolicyIDs* attribute, either explicitly (setting the attribute in the resource itself) or implicitly (either by using the parent privileges or the system default policies). Which means that the system shall provide a default access privileges in case that the Originator does not provide a specific *accessControlPolicyIDs* during the creation of the resource.To update this attribute, a Hosting CSE shall check whether an Originator has Update permission in any *selfPrivileges* of the *<accessControlPolicy>* resources which this attribute originally indicates. |

### But this information is missing in description of selfPrivileges in section 9.6.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *selfPrivileges* | 1 | RW | A set of access control rules that apply to the *<accessControlPolicy>* resource itself. | MA |

### So this CR proposes to modify the description of selfPrivileges attribute in section 9.6.2 to give the correct understanding.

### -----------------------Start of change 1----------------------------------------------

### 9.6.2 Resource Type *accessControlPolicy*

#### 9.6.2.0 Introduction

The Access Control Policies (ACPs) shall be used by the CSE to control access to the resources as specified in the present document and in oneM2M TS-0003 [2].

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

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

In a privilege, each access control rule defines which AE/CSE is allowed for which operation. So for sets of access control rules an operation is permitted if it is permitted by one or more access control rules in the set.

For a resource that is not of *<accessControlPolicy>* resource type, the common attribute *accessControlPolicyIDs* for such resources (defined in table 9.6.1.3.2-1) contains a list of identifiers which link that resource to *<accessControlPolicy>* resources. The CSE Access Decision for such a resource shall follow the evaluation of the set of access control rules expressed by the *privileges* attributes defined in the *<accessControlPolicy>* resources.

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

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

 Logically an authorization system may comprise four sub-functions: enforcing access control decision, making access control decision, providing access control policies and providing access control information (e.g. roles). As specified in TS-0003 [2], these sub-functions are modelled as policy enforcement point (PDP), Policy Decision Point (PDP), Policy Retrieval Point (PRP) and Policy Information Point (PIP) respectively. In oneM2M System these authorization sub-functions may coexist in one CSE or may be distributed in different CSEs in different combinations.

In *<accessControlPolicy>* resource three operational attributes are defined for holding the information about where to find the distributed authorization sub-functions. These attributes are: *pdpResourceIDs*, *prpResourceIDs* and *pipResourceIDs*.

The *pdpResourceIDs* attribute contains a list of addresses of <*authorizationDecision*> resources. Each <*authorizationDecision*> resource represents a PDP to which an access control decision request shall be sent in order to obtain an access control decision. See clause 9.6.42 for further details of <*authorizationDecision*> resource type.

The *prpResourceIDs* attribute contains a list of addresses of <*authorizationPolicy*> resources. Each <*authorizationPolicy*> resource represents a PRP to which an access control policy request shall be sent in order to obtain access control policies. See clause 9.6.43 for further details of <*authorizationPolicy*> resource type.

The *pipResourceIDs* attribute contains a list of addresses of <*authorizationInformation*> resources. Each <*authorizationInformation*> resource represents a PIP to which an access control information request shall be sent in order to obtain requested access control information (e.g. role and/or token) for making an access control decision. See clause 9.6.44 for further details of <*authorizationInformation*> resource type.

When there is a resource associated with more than one authorization schemes, the order to apply the schemes shall follow static/non-distributed (i.e. applying *accessControlPolicyIDs* attribute of a target resource and *privileges* attribute of a <accessControlPolicy> resource), dynamic authorization consultation (i.e. applying *dynamicAuthorizationConsultationIDs* attribute of a target resource) and then distributed (i.e. *authorizationDecisionResourceIDs*, *authorizationPolicyResourceIDs* and *authorizationInformationResourceIDs* attributes of a <accessControlPolicy> resource). The Hosting CSE shall try the next scheme when it fails to find the privilege to grant the Originator’s request.

Editor’s Note: the name “static/non-distributed” is a temporal terminology so it would be refined

Editor’s Note: the detailed mechanism to apply the three different authorization scheme will be specified in clause 10

The applicability of the *authorizationDecisionResourceIDs*, *authorizationPolicyResourceIDs* and *authorizationInformationResourceIDs* attributes for the distributed authorization depends on the deployment form of authorization sub-functions:

* In the case the *privileges* attribute is not NULL, the access control rules in the *privileges* attribute shall be used for access control, and the *authorizationDecisionResourceIDs*, *authorizationPolicyResourceIDs* and *authorizationInformationResourceIDs* attributes shall not be present.
* In the case the *privileges* attribute is NULL, how to process further depends on which authorization method is adopted. In the case distributed authorization method is supported, *authorizationDecisionResourceIDs* or *authorizationPolicyResourceIDs* attribute shall be considered for obtaining access control decision or access control policies from another CSE. However, *authorizationDecisionResourceIDs* and *authorizationPolicyResourceIDs* attributes shall not be present at the same time.
* In case the *authorizationInformationResourceIDs* attribute is present, the access control information request (e.g. for role information) related to the access control policy specified in the *privileges* attribute shall be sent to one of the addresses listed in this attribute.

The details of distributed authorization procedures are described in TS-0003 [2].



Figure 9.6.2.0-1: Structure of *<accessControlPolicy>* resource

The *<accessControlPolicy>* resource shall contain the child resource specified in table 9.6.2.0-1.

Table 9.6.2.0-1: Child resources of *<accessControlPolicy>* resource

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

The *<accessControlPolicy>* resource shall contain the attributes specified in table 9.6.2.0-2.

Table 9.6.2.0-2: Attributes of *<accessControlPolicy>* resource

| Attributes of *<accessControlPolicy>* | Multiplicity | RW/RO/WO | Description | *<accessControlPolicyAnnc>* Attributes |
| --- | --- | --- | --- | --- |
| *resourceType*  | 1 | RO | See clause 9.6.1.3. | NA |
| *resourceID* | 1 | RO | See clause 9.6.1.3. | NA |
| *resourceName* | 1 | WO | See clause 9.6.1.3. | NA |
| *parentID* | 1 | RO | See clause 9.6.1.3. | NA |
| *expirationTime* | 1 | RW | See clause 9.6.1.3. | MA |
| *labels* | 0..1(L) | RW | See clause 9.6.1.3. | MA |
| *creationTime* | 1 | RO | See clause 9.6.1.3. | NA |
| *lastModifiedTime* | 1 | RO | See clause 9.6.1.3. | NA |
| *announceTo* | 0..1 (L) | RW | See clause 9.6.1.3. | NA |
| *announcedAttribute* | 0..1 (L) | RW | See clause 9.6.1.3. | NA |
| *privileges* | 1 | RW | A set of access control rules that applies to resources referencing this *<accessControlPolicy>* resource using the *accessControlPolicyID* attribute. | MA |
| *selfPrivileges* | 1 | RW | A set of access control rules that apply to the *<accessControlPolicy>* resource itself and *accessControlPolicyIDs* attribute of any other resource which is linked to this <accessControlPolicy> resource. | MA |
| *authorizationDecisionResourceIDs* | 0..1 (L) | RW | A list of addresses of <*authorizationDecision*> resources. See clause 9.6.42 for further details. | MA |
| *authorizationPolicyResourceIDs* | 0..1 (L) | RW | A list of addresses of <*authorizationPolicy*> resources. See clause 9.6.43 for further details. | MA |
| *authorizationInformationResourceIDs* | 0..1 (L) | RW | A list of addresses of <*authorizationInformation*> resources. See clause 9.6.44 for further details. | MA |

The set of access control rules represented in *privileges* and *selfPrivileges* attributes are comprised of 4-tuples (*accessControlOriginators*, *accessControlContexts*, *accessControlOperations, accessControlObjectDetails*) with parameters shown in table 9.6.2.0-3 which are further described in the following clauses.

If the *privileges* attribute contains no 4-tuples then this represents an empty set of the access control rules.

The *selfPrivileges* attribute shall contain at least one tuple.

The CSE access granting mechanism shall follow the procedure described in oneM2M TS-0003 [2] in clause 7.1 (Access Control Mechanism).

Table 9.6.2.0-3: Parameters in access-control-rule-tuples

| Name | Description |
| --- | --- |
| *accessControlOriginators* | See clause 9.6.2.1 |
| *accessControlContexts* | See clause 9.6.2.2 |
| *accessControlOperations* | See clause 9.6.2.3 |
| *accessControlObjectDetails* | See clause 9.6.2.4 |
| *accessControlAuthenticationFlag* | See clause 9.6.2.5 |

#### 9.6.2.1 *accessControlOriginators*

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

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

| Name | Description |
| --- | --- |
| *domain* | A SP domain or SP sub-domain |
| *originatorID* | CSE-ID, AE-ID or the resource-ID of a <group> resource that contains the AE or CSE that represents the Originator. |
| *all* | Any Originators are allowed to access the resource within the *accessControlOriginators* constraints |
| *Role-ID* | A Role Identifier as defined in clause 7.1.14 |

When the *originatorID* is the resource-ID of a <group> resource which contains <AE> or <remoteCSE> as member, the Hosting CSE of the resource shall check if the originator of the request matches one of the members in the memberIDs attribute of the <group> resource (e.g. by retrieving the <group> resource). If the <group> resource cannot be retrieved or doesn't exist, the request shall be rejected.

#### 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. |
| *accessControlIpIPAddress* | Represents an IP address constraint or IP address block constraint which is compared against the IP address of the Originator of the request. |

#### 9.6.2.3 *accessControlOperations*

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

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

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

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

#### 9.6.2.4 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.

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  |
| *specialization* | 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*. |

#### 9.6.2.5 *accessControlAuthenticationFlag*

The *accessControlAuthenticationFlag* is an optional parameter in an access-control-rule-tuple: if the value is TRUE, then the access control rule applies only if the Originator is considered to be authenticated by the Hosting CSE; if the value is FALSE, then the access control rule applies whether or not the Originator is considered to be authenticated by the Hosting CSE. Clause 7.1.2 in oneM2M TS-0003 [2] describes the criteria used to determine if the Originator is considered to be authenticated by the Hosting CSE.

If the *accessControlAuthenticationFlag* parameter is not present, then the value is assumed to be FALSE.

### -----------------------End of change 1----------------------------------------------

CHECK LIST

* Does this change request include an informative introduction containing the problem(s) being solved, and a summary list of proposals.?
* Does this CR contain changes related to only one particular issue/problem?
* Have any mirror crs been posted?
* Does this change request make **all** the changes necessary to address the issue or problem? 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?
* Does this change request follow the drafting rules?
* Are all pictures editable?
* Have you checked the spelling and grammar?
* Have you used change bars for all modifications?
* Does the change 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 sections need not show surrounding clauses as long as the proposed section number clearly shows where the new section is proposed to be located.)
* Are multiple changes in this CR 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.?