|  |
| --- |
|  |

|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | SDS #68 |
| Source:\* | Mohd Uvaish Siddiqui, C-DOT, uvaish@cdot.in  Prateek Varshney, C-DOT, prateekv@cdot.in  Poornima Shandilya, C-DOT, poornima@cdot.in  Anupama Chopra, C-DOT, anupama@cdot.in |
| Date:\* | 2025-02-13 |
| Reason for Change/s:\* | TS-0003 – accessControlObjectDetails handling in <accessControlPolicy> resource |
| 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-0003 v3.17.1 |
| Clauses \* | 7.1.5 |
| 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) | TS-0001 |
| 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

In TS-0001 and TS-0004 for <mgmtObj> or <flexContainer> specializations, specializationType attribute is used while in TS-0003 specializationID or specializationType both words are used causing the inconsistency in the specs. Thus, the CR proposes to rename the specializationID parameter of accessControlObjectDetails of <accessControlPolicy> resource to specializationType for Release 3 and Release 4 of TS-0003 to maintain consistency.

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

### 7.1.5 Description of the Access Decision Algorithm

The reference access decision algorithm specified in this clause combines partial access control results obtained for each of the individual access control rules contained in a *privileges* or *selfPrivileges* attribute. Further, if multiple ACP instances are assigned to the protected resource, the reference access decision algorithm combines the partial access control results obtained for the individual ACPs of an ACP set.

The algorithm specified in this clause adopts a "Permit-overrides" combining algorithm with respect to access control rules and ACPs as defined in XACML [i.5]. This algorithm has the following behaviour:

1. If a decision is "Permit" for only a single access control rule included in the *privileges* (or *selfPrivileges*) attribute of a single ACP, the result is "Permit".
2. Otherwise, the result is "Deny".

The logic for evaluating a request against a privilege can be described mathematically as follows. A *privileges* or *selfPrivileges* attribute included in an <*accessControlPolicy*> resource represents a set of access control rules, *acrs*, which is built as in figure 7.1.5-1.



Figure 7.1.5-1: Logic to evaluate privileges in the reference access decision algorithm

The parameters associated with a request, which are evaluated against the parameters contained in the access control rules are specified in clause 7.1.3.

The access decision *res\_acrs* defined in clause 7.1.4 is derived by evaluating whether or not the parameters associated with the request message listed in tables 7.1.2-1 and 7.1.2-2 match any of the access control rules contained in the access control rule set defined in clause 7.1.3 as follows:

*res\_acrs* = *res\_acr*(1) OR *res\_acr*(2) ... OR *res\_acr*(k) … OR *res\_acr*(K),

where *res\_acr*(*k*) represents the logical evaluation result (i.e. TRUE/FALSE or 1/0) of the request parameters against the *k*th access control rule in the set *acrs*, which can be expressed as follows:

*res\_acr*(*k*) = *res\_authn(k)* AND *res\_origs*(*k*) AND *res\_ops*(*k*) AND *res\_ctxts*(*k*) AND *res\_objd*(*k*), *k* = 1…K.

The first partial logical result variable *res\_authn(k)* on the right side of above equation shall be evaluated according to Table 7.1.5-1:

Table 7.1.5-1: Evaluating *res\_authn(k)*

| *acr(k)\_*accessControlAuthenticationFlag | *rq\_authn* | *res\_authn* |
| --- | --- | --- |
| TRUE | TRUE | TRUE |
| TRUE | FALSE | FALSE |
| FALSE | TRUE | TRUE |
| FALSE | FALSE | TRUE |

The remaining 4 partial logical result variables on the right side of above equation can be defined by using the following set function:



With this definition:

*res\_origs*(*k*) = ismember(***Originator***, *acr*(*k*)\_accessControlOriginators)

*res\_ops*(*k*) = ismember(***Operation***, acr(*k*)\_ accessControlOperations)

In the above equation, the ***Originator*** variable refers to the authenticated identity of the originator of the request primitive which matches the ***From*** parameter.

The third partial logical result *res\_ctxts*(*k*) is derived as follows:

*res\_ctxts*(*k*) = *res\_context*(*k*, 1) ... OR *res\_context*(*k*, *m*) ... OR *res\_context*(*k,* M\_*k*),

where:

*res\_context*(*k*, *m*) = *res\_time*(*k*, *m*) AND *res\_ip*(*k*, *m*) AND *res\_loc* (*k*, *m*), k = 1…K, *m* = 1…M\_*k*

and

*res\_time*(*k*, *m*) = ismember(***rq\_time***, *acr*(*k*)\_accessControlWindow(*m*))

*res\_ip*(*k*, *m*) = ismember(***rq\_ip***, *acr*(*k*)\_accessControlIpAddresses(*m*))

*res\_loc* (*k*, *m*) = ismember(***rq\_loc***, *acr*(*k*)\_accessControlLocationRegion(*m*))

The fourth partial logical result *res\_objd*(*k*) applies to Create request primitives only and is derived as

*res\_ objd*(*k*) = *res\_ objdetails*(*k*, 1) ... OR *res\_ objdetails*(*k*, *m*) ... OR *res\_ objdetails*(*k,* M\_*k*),

where:

*res\_ objdetails*(*k, m*) = *res\_resourceType*(*k, m*) AND *res\_specializationType*(*k, m*) AND *res\_childResource*(*k,m*),

for *m* = 1…M\_*k***.** The three logical arguments are defined below.

For each given element *acr*(*k*)\_accessControlObjectDetails(*m*) in an access control rule determine if the optional *resourceType* parameter is present

*resourceType* = *acr*(*k*)\_accessControlObjectDetails(*m*)/resourceType

Depending on the presence of *resourceType*, *res\_resourceType*(*k, m*) is derived as



where *targetResourceTypeID* is the resource type identifier associated with the resource addressed in the ***To*** parameter of the Create request primitive.

If the value of the *resourceType* element is 13 (<mgmtObject> specialization) or 28 (<flexContainer> specialization>), the optional specializationType element shall also be included in accessControlObjectDetails:

*specializationType* = *acr*(*k*)\_accessControlObjectDetails(*m*)/specializationType

If *specializationType* is present, it shall be matched against the *mgmtDefinition* or *containerDefinition* attributes given in the ***Content*** parameter of the Create request primitive.

The *childResourceType* element is mandatory in any given accessControlObjectDetails element of an access control rule. It includes a list of *j* = 1…J child resource type identifiers to which the rule applies. The jth list element is denoted as follows

*childResourceType*(*k*, *m*. *j*) = *acr*(*k*)\_accessControlObjectDetails(*m*)/childResourceType(*j*), *j* = 1…J

The logical variable *res\_childResource*(*k, m*) is derived as

*res\_ childResource* (*k, m*) = ismember(***Resource Type***, *childResourceType*(*k*, *m*, *j*))

where ***Resource Type*** refers to the value of the parameter of the given Create request primitive.

NOTE: If resourceType and specializationType are not present in acr(k)\_accessControlObjectDetails(m), res\_ objdetails(k, m) = res\_resourceType(k, m) AND res\_specializationType(k, m) AND res\_childResource(k,m) = res\_childResource(k,m).

Thanks to the "Permit-overrides" combining approach, if the access control decision for one access control rule results in *res\_acr* = TRUE, the reference access decision algorithm can stop without evaluating any other applicable access control rules of the current ACP or any other ACPs in the ACP set, and the final access decision is "Permit".

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