|  |
| --- |
| Input Contribution |
| Meeting ID\* | SDS 39 |
| Title:\* | Attribute-level access control |
| Source:\* | Dale Seed, Convida, Seed.Dale@convidawireless.com Jiwan Ninglekhu, Convida, Ninglekhu.jiwan@convidawireless.com |
| Date:\* | 2018-02-10 |
| Input related to\* | WI-0077 |
| Intended purpose ofdocument:\* | [x]  Decision[ ]  Discussion[ ]  Information[ ]  Other <specify> |
| Impacted other TS/TR(s) | TR-0050-v0.6.0 |
| Decision requested or recommendation:\* | Approval of the description of attribute-level access control |
| 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.

#

# Introduction

oneM2M employs access control policies and mechanisms that can control an originator’s access to the level of granularity of an individual resource. This contribution proposes a solution for Attribute-Level Access Control. It defines an enhancement to the existing oneM2M access control policy mechanism to control access down to the level of granularity of an individal attribute of a resource.

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

### 6.3.2 Solution #1.1.2: Attribute-Level Access Control

#### 6.3.2.1 Introduction

oneM2M access control policies define access privileges for oneM2M resources. Currently, the lowest level of granularity of privileges supported are resource level privileges. Resource level privileges define which entities are allowed to access a resource and the operations they are allowed to perform on the entire resource. The following solution proposes to add further granularity to support attribute level privileges. Attribute level privileges define the entities that are allowed to access individual attribute(s) of a resource and the allowed operations they are permitted to perform on these individual attribute(s).

#### 6.3.2.2 Solution details

6.3.2.2.1 Attribute-Level Access Control Rules

Clause 7.1.3 of TS-0003[i.2] defines the *privileges* and *selfPrivileges* attributes of the <*accessControlPolicy*> resource as a set of access control rules. The set of access control rules is denoted as *acrs* and an individual access control rule in this set is denoted as an *acr*. The individual access control rules in *acrs* are indexed with the letter *k*. The number of access control rules in the set is denoted with the letter K.

*acrs* = { *acr*(1), *acr*(2), ..., *acr*(*k*), ..., *acr*(K) }

Currently, each access control rule *acr*(*k*) is comprised of five types of access-control-rule-tuple parameters, denoted as accessControlOriginators, accessControlOperations, accessControlContexts, accessControlObjectDetails and accessControlAuthenticationFlag.

To support attribute level access, a sixth access-control-rule-tuple parameter is defined and is denoted as accessControlAttributes. The definition of accessControlAttributes is shown in Table 6.3.2.2.1-1.

Table 6.3.2.2.1-1: Additional parameters of an access-control-rule-tuple

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Usage Description | Mandatory/Optional | Format |
| accessControlAttributes | Set of resource attributes for which access can be authorized  | O | List of resource attribute name(s).  |

The accessControlAttributes parameter comprises a list of accessible resource attributes names. The list includes one or more names of oneM2M resource attributes represented in their short name format as defined in oneM2M TS-0004[i.4].

The data type applicable to accessControlAttributes will be defined in oneM2M TS-0004[i.4]. A proposed type is m2m:attributeList but this is FFS.

6.3.2.2.2 Access Control Decision

Figure 6.3.2.2.2-1 shows the modifications to the access decision algorithm defined in TS-0003[i.2] required to support the accessControlAttributes access-control-rule-tuple parameter.



Figure 6.3.2.2.2-1: Logic to evaluate privilege in the reference access decision algorithm

Clauses 7.1.4 and 7.1.5 of TS-0003[i.2] define *res\_acrs* 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 of *acrs.* The modification to *res\_acr*(*k*) required to support the accessControlAttributes access-control-rule-tuple parameter is 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*) AND *res\_attrs*(*k*),

where *k* = 1…K, and

*res\_attrs*(*k*) = ismember(***rq\_attributes****, acr*(*k*)*\_*accessControlAttributes),

where ***rq\_attributes*** refers to the targeted attributes specified in the ***To*** or ***Content*** parameter of the request.

If all the requested attribute names referenced by ***rq\_attributes*** match the names of attributes present in  *acr*(*k*)*\_accessControlAttributes*, then *res\_attrs*(*k*) is True or 1, otherwise *res\_attrs*(*k*) is False or 0.

Note, attribute level access control checks are optional and only performed for access-control-rule-tuples that include an accessControlAttributes parameter. If an access-control-rule-tuple does not include an accessControlAttributes parameter, then only resource level access control checks are performed for that rule.

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