|  |
| --- |
|  |

|  |  |
| --- | --- |
| 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 4 |
| 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 v4.8.1 |
| 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 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 7.1.3 Format of *privileges* and *selfPrivileges* Attributes

The *privileges* and *selfPrivileges* attributes exhibit the same data type format which is specified as follows.

Each *privileges* or *selfPrivileges* attribute comprises a set of access control rules. In the following, the set of access control rules is denoted as *acrs* and an individual access control rule in this set as *acr*. The 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) }

Each access control rule *acr*(*k*) is comprised of mandatory accessControlOriginators and accessControlOperations components and optional accessControlContexts, accessControlObjectDetails, accessControlAuthenticationFlag and accessControlAttributes components.

Hence, an access control rule *acr*(*k*) is either represented as a pair:

*acr*(*k*) = {*acr*(*k*)\_accessControlOriginators, *acr*(*k*)\_accessControlOperations}

or as a 3-tuple, 4-tuple, 5-tuple or 6-tuple. For example, a 3-tuple such as the following:

*acr*(*k*) = {*acr*(*k*)\_accessControlOriginators, *acr*(*k*)\_accessControlOperations, *acr*(*k*)\_accessControlContexts}

The generic term "access-control-rule-tuple" is used when referring to a rule *acr*(*k*).

A set *acrs* of access control rules may consist of a mix of pairs,3-tuples, 4-tuples, 5-tuples or 6-tuples. For pairs or for any tuples not containing accessControlContexts, any context parameters associated with a request message are admissible.

The six component parameters of an access-control-rule-tuple supported in the present document are shown in table 7.1.3-1.

Table 7.1.3-1: Parameters of an access-control-rule-tuple

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Usage Description | Mandatory/Optional | Format |
| accessControlOriginators | Set of Originators that can be authorized | M | List of CSE-IDs and/or AE‑IDs, or keyword "all" to grant access to all originators |
| accessControlOperations | Set of Operations that can be authorized | M | Enumerated list of operations Create Retrieve, Update, Delete, Discover, Notify |
| accessControlContexts | See table 7.1.3-3 | O | See table 7.1.3-3 |
| accessControlObjectDetails | See table 7.1.3-4 | O | See table 7.1.3-4 |
| accessControlAuthenticationFlag | Indicates whether the rule applies only to Originators which are considered to be authenticated by the Hosting CSE | O | Boolean |
| accessControlAttributes | Set of resource attributes for which access can be authorized | O | List of resource attribute name(s). |

The accessControlOriginators parameter comprises a list of SP domain names, CSE-IDs, AE-IDs, resource-IDs of <group> resources and/or Role IDs of any format defined in oneM2M TS‑0001 [1]. If access for all originators is to be allowed, the reserved keyword "all" may be included into the value space of accessControlOriginators.

Using a SP domain name in accessControlOriginators means all AE-IDs and CSE-IDs matching the given domain name can be authorized.

It is furthermore allowed to use wildcard character "\*", in representations of M2M-SP-ID (i.e., SP domain names), CSE-ID and AE‑ID. The scope of a "\*" is terminated by a following "/" character. Table 7.1.3-2 shows examples of using wildcard characters in CSE-IDs and AE-IDs.

Wildcard characters are not permitted in resource-IDs of <group> resources and Role IDs.

Table 7.1.3-2: Examples of using wildcard characters in CSE-IDs and AE-IDs of accessControlOriginators

|  |  |  |  |
| --- | --- | --- | --- |
|  | Form of ID | Examples | Meaning |
| CSE-ID | Absolute | //m2msp.org/myCSEID  //\*/myCSEID  //\*/myCSE\* | Any CSE whose ID matches the wild cards |
|  | SP-relative | /myCSEID  /myCSE\* | Any matching CSE from the SP that is hosting the target resource |
| AE-ID | Absolute | //m2msp.org/S988  //\*/myCSEID/C9886  //\*/myCSE\*/C9886 | Any AE whose ID matches the wild cards |
|  | SP-relative | /myCSEID/C9886  /myCSEID/C98\*  /myCSE\*/C98\*  /SmyAE\* | Any matching AE from the SP that is hosting the target resource |

The data type applicable to accessControlOriginators is defined in oneM2M TS-0004 [4].

The accessControlOperations parameter comprises a list of admissible operations which can be any subset of the following elements: Create, Retrieve, Update, Delete, Discover, and Notify. While Create, Retrieve, Update, Delete, and Notify operation are explicitly indicated in the *op* parameter of a request message, the Discovery operation is indicated by the *filterUsage* condition of the ***Filter Criteria*** request parameter having a value of “Discovery”, “Discovery-based Operation” or “IPE On-Demand Discovery”.

The data type applicable to accessControlOperations is defined in oneM2M TS-0004 [4].

The accessControlContexts parameters are listed in table 7.1.3-3.

Table 7.1.3-3: Parameters of accessControlContexts

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Usage Description | Mandatory/Optional | Formats |
| accessControlTimeWindow | Set of Time Windows that can be authorized | O | List of time intervals where access can be granted in extended crontab format |
| accessControlLocationRegion | Set of Location Regions that can be authorized | O | 1) Latitude/longitude coordinates, and a radius defining a circular region around the coordinates  2) Country code |
| accessControlIpAddress | Set of IPv4 and IPv6 addresses that can be authorized | O | IPv4: dotted-decimal notation with CIDR suffix  IPv6: colon separated groups of hexadecimal digits with CIDR suffix |
| accessControlUserIDs | Set of M2M Service Users that can be authorized | O | List of M2M-User-IDs |
| accessControlEvalCriteria | Set of conditions that are factored into authorization decisions. | O | A tuple consisting of a mandatory resource identifier of a subject resource and an set of evaluation criteria applicable to the subject resource. |
| accessControlLimit | Number of times access to a resource can be authorized. | O | A number that indicates how many times access can be granted. |

The accessControlTimeWindow parameter represents a list of elements that comply with the extended crontab syntax as defined in clause 7.3.8 of oneM2M TS-0004 [4]. It allows definition of periodically recurring time intervals at which access can be granted, when the ***rq\_time*** parameter associated with the access request message falls into such interval.

For the elements of accessControlLocationRegion there are two representation choices. These can be represented by a 2‑character country code or a circle with radius *R* centred at a point defined in terms of longitude and latitude parameters. Refer to Annex F for detailed information. Each element of accessControlLocationRegion defines an admissible location region, which is compared with the ***rq\_loc*** parameter associated with the access request message.

The data types applicable to accessControlLocationRegion and ***rq\_loc*** are defined in oneM2M TS-0004 [4].

The accessControlIpAddress parameter represents a list of IPv4 and IPv6 addresses in dotted-decimal notation with CIDR suffix or colon separated groups of hexadecimal digits with CIDR suffix, respectively. If the ***rq\_loc*** parameter associated with the access request message matches one of these addresses, access may be granted with regard to this criterion.

The data types applicable to accessControlIpAddress and ***rq\_ip*** are defined in oneM2M TS-0004 [4].

The accessControlUserIDs parameter comprises a list of M2M-User-IDs having a format defined in oneM2M TS‑0001 [1]. Using just a SP domain name in accessControlUserIDs means all M2M-User-IDs matching the given domain name can be authorized. For example, “//m2msp.org”. It is furthermore allowed to use a wildcard character "\*" within the SP-Relative-M2M-User-ID portion of a M2M-User-ID. For example, //m2msp.org/homeowner\*. A wildcard character is not permitted within the SP domain name portion of a M2M-User-ID.

The data type applicable to accessControlUserIDs is defined in oneM2M TS-0004 [4].

This accessControlEvalCriteria parameter represents the conditions determining if the request operation is to be allowed. It allows conditional access to the resource based on conditions not contained in the received request. The accessControlEvalCriteria parameter is a tuple that consists of a mandatory subjectResourceID element as defined in table 9.6.61-2 in oneM2M TS‑0001 [1] and an evalCriteria element defined in table 9.6.61-3 in oneM2M TS-0001 [1].

The accessControlLimit parameter represents the number of times that the policy defined by an access-control-rule-tuple can allow authorization to the requested resource. This attribute maintains the number of remaining accesses allowed. The parameter is decremented each time an access to the requested resource is granted. If the value is greater than zero then access is granted, otherwise access is denied. If the accessControlLimit parameter is not present in an access-control-rule-tuple, then there are no restrictions on the number of times access is granted.

The accessControlAuthenticationFlag parameter is a Boolean value. If the accessControlAuthenticationFlag parameter is not present, then the value is assumed to be FALSE. If the accessControlAuthenticationFlag parameter is TRUE, then this indicates that the access control rule applies only to Originators considered to have been authenticated by the Hosting CSE. Clause 7.1.2 specifies the criteria used to decide whether or not the Originator is considered to have been authenticated by the Hosting CSE.

The accessControlObjectDetails parameters are listed in table 7.1.3-4.

Table 7.1.3-4: Parameters of accessControlObjectDetails

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Usage Description | Mandatory/Optional | Formats |
| resourceType | Resource type on which access control rule applies | O | Resource type identifier |
| specializationType | Identifier of mgmtDefinition or containerDefinition | O | mgmtDefinition or containerDefinition represented as a string. |
| childResourceType | Set of resource type identifiers that can be created under the parent resource. | O | Resource type list. |

The accessControlObjectDetails attribute 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* is specified. An access control rule which does not include any *accessControlObjectDetails* parameters applies to all child resource types of the target resource. The *accessControlObjectDetails* parameter is described in table 9.6.2.4-1 of oneM2M TS‑0001 [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* elements are optional. If either the *resourceType* or *specializationType* element is present in *accessControlObjectDetails*, the CSE matches the type of resource or specialization of the targeted resource with the value specified in the *resourceType* or *specializationType* element. Further checking of *childResourceType* is done only if the *resourceType* or *specializationType* match occurs. However, if the *resourceType* and *specializationType* elements are not provided, then only *childResourceType* match is performed.

The accessControlAttributes attribute specifies a list of one or more resource attribute names . If there is a rule for which all conditions of the rule are satisfied, then other rules shall be ignored. Otherwise, rules that contain *accessControlAttributes* and that satisfy all conditions apart from *accessControlAttributes* are considered to be applicable rules. In this case, the resource attributes associated with the request and its response are evaluated against the union of resource attributes defined across all the *accessControlAttributes* of these applicable rules to determine if access is allowed.

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

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

### 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. The first phase of the algorithm determines if the result is “Permit” if a single access control rule included in the *privileges* (or *selfPrivileges*) attribute of a single ACP permits access to the targeted resource.
2. The second phase of the algorithm determines if the result is “Permit” if the set of applicable access control rules containing *accessControlAttributes* collectively as a union permit access to the targeted resource.
3. 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*) AND *res\_attrs*(*k*),

where *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 next 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 fourth 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*) AND *res\_uids(k, m)*, k = 1…K, *m* = 1…M\_*k*

and

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

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

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

*res\_uids*(*k*, *m*) = ismember(***M2M Service User***, *acr*(*k*)\_accessControlUserIDs(*m*))

The fifth 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).

The sixth partial logical result *res\_attrs*(*k*) is derived as follows:



Depending on the type of operation, the requested attribute names defined within the parameters of the request (e.g., ***To, Content,*** ***Filter Criteria***) or within the targeted resource shall be compared against the names of attributes present in *acr(k)\_*accessControlAttributes to determine the value of *res\_attrs(k)* as follows:

* For an operation that includes a ***Filter Criteria*** parameter and that requires access to the attributes of a resource to process the ***Filter Criteria*** (i.e., matching conditions defined within a discovery operation, discovery-based operation, IPE On-demand discovery operation or a conditional operation), *acr(k)\_*accessControlAttributes defines the attributes that can be accessed. If the ***Filter Criteria*** includes the names of attributes that are not defined in *acr(k)\_*accessControlAttributes, then then *res\_attrs*(*k*) is False or 0. Otherwise, if the names of all the attributes are defined in *acr(k)\_*accessControlAttributes, then the value of *res\_attrs*(*k*) shall be determined by the operation specific steps described below:
* For a Retrieve operation, *acr(k)\_*accessControlAttributes defines the attributes that can be retrieved and included in the response.
  + For a Retrieve operation that targets a resource, in which all the names of the attributes present in the targeted resource are included in *acr(k)\_*accessControlAttributes, then *res\_attrs*(*k*) is True or 1. Otherwise, if one or more of the names of the attributes present in the targeted resource are not included in *acr(k)\_*accessControlAttributes, then *res\_attrs*(*k*) is False or 0.
  + For a Retrieve operation that targets one or more individual attributes of a resource (i.e., partial retrieve) and these attributes are all defined in *acr(k)\_*accessControlAttributes, then *res\_attrs*(*k*) is True or 1. Otherwise, if one or more individual attributes are not defined in *acr(k)\_*accessControlAttributes, then *res\_attrs*(*k*) is False or 0.
* For a Delete operation, *acr(k)\_*accessControlAttributes defines the attributes that can be deleted. If all the attributes present in the targeted resource of a Delete operation are defined in *acr(k)\_*accessControlAttributes, then *res\_attrs*(*k*) is True or 1. Otherwise, if one or more of the attributes are not defined in *acr(k)\_*accessControlAttributes, then *res\_attrs*(*k*) is False or 0.
* For an Update operation, *acr(k)\_*accessControlAttributes defines the attributes that can be included in the ***Content*** parameter of a request and its response. For an Update operation that attempts to create, update or delete one or more attributes of a resource that are all defined in *acr(k)\_*accessControlAttributes, then *res\_attrs*(*k*) is True or 1, however any attributes of the targeted resource not included in *acr(k)\_*accessControlAttributes shall be filtered and not included in the response. Otherwise, if one or more of the attributes of the Update operation are not defined in *acr(k)\_*accessControlAttributes, then *res\_attrs*(*k*) is False or 0.
* For a Create operation, *acr(k)\_*accessControlAttributes defines the attributes that can be included in the ***Content*** parameter of a request and its response. For a Create operation that attempts to create a resource with attributes that are all defined in *acr(k)\_*accessControlAttributes, then *res\_attrs*(*k*) is True or 1, however any attributes of the targeted resource not included in *acr(k)\_*accessControlAttributes shall be filtered and not included in the response. Otherwise, if one or more attributes of the Create operation are not defined in *acr(k)\_*accessControlAttributes, then *res\_attrs*(*k*) is False or 0.

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" (i.e. *res\_acrs* = TRUE).

However, if the first phase of the reference access decision algorithm results in *res\_acrs* = FALSE, and during the processing of the algorithm, one or more access-control-rule-tuple including an *accessControlAttributes* condition is processed, then a second phase of the access decision algorithm shall determine the final access decision. In the second phase of the access decision algorithm, the following steps shall be performed:

* all access control rules having *accessControlAttributes* conditions, that have satisfied all conditions of the access decision algorithm apart from their *accessControlAttributes* condition, shall be collectively considered an applicable set of access control rules,
* Depending on the type of operation, the requested attribute names defined within the parameters of the request (e.g., ***To, Content,*** ***Filter Criteria***) or within the targeted resource shall be compared against the names of attributes present in the union of resource attributes defined across all the *accessControlAttributes* of the applicable set of access control rules to determine the value of *res\_acrs* as follows:
  + If a Retrieve, Delete, Update or Create operation includes a ***Filter Criteria*** parameter with names of one or more attributes that are not defined in the union of *accessControlAttributes*, then the final access decision shall be "Deny". Otherwise, if the names of the attributes are all defined in the union of *accessControlAttributes*, then the final access decision shall be determined by the operation specific steps described below:
  + For a Retrieve operation that targets a resource, the final access decision shall be "Permit", but any attributes not included in the union of *accessControlAttributes* shall be filtered and not included in the response. If none of the attributes defined in the union of *accessControlAttributes* match the names of the attributes present in the targeted resource, then no attributes shall be returned in the response.

For a Retrieve operation that targets one or more individual attributes of a resource (i.e., partial retrieve) and these attributes are all defined in the union of *accessControlAttributes*, then the final access decision shall be "Permit". Otherwise, if one or more individual attributes are not defined in the union of *accessControlAttributes*, then the final access decision shall be "Deny".

* + For a Delete operation, if all the attributes present in the targeted resource are defined in the union of *accessControlAttributes*, then the final access decision shall be "Permit". Otherwise, if one or more of the attributes present in the targeted resource are not defined in union of *accessControlAttributes*, then the final access decision shall be "Deny".
  + For an Update operation that attempts to create, update or delete one or more attributes of a resource that are all defined in the union of *accessControlAttributes*, then the final access decision shall be "Permit", however any attributes of the targeted resource not included in the union of *accessControlAttributes* shall be filtered and not included in the response. Otherwise, if one or more of the attributes of the attempted Update operation are not defined in the union of *accessControlAttributes*, then the final access decision shall be "Deny".
  + For a Create operation that attempts to create a resource with attributes that are all defined in the union of *accessControlAttributes*, then the final access decision shall be "Permit", however any attributes of the targeted resource not included in the union of *accessControlAttributes* shall be filtered and not included in the response. Otherwise, if one or more attributes of the attempted Create operation are not defined in the union of *accessControlAttributes*, then the final access decision shall be "Deny".

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