|  |
| --- |
|  |

|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | ARC 29 |
| Source:\* | Francisco Sang-Eon Kim, KT (TTA), kim.sangeon@kt.com |
| Date:\* | 2017-05-14 |
| Reason for Change/s:\* | Theaccess mode of *memberType* attribute of <*group*> resource has problem considering *consistencyStrategy* attribute. |
| CR against: Release\* | Release 3 |
| CR against: WI\* | Active <Work Item number>  MNT maintenance  Is this a mirror CR? Yes  No  mirror CR number: (Note to Rapporteur - use latest agreed revision)  STE Small Technical Enhancements / WI-0050  Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0001-V3.5.0 |
| Clauses \* | * + 1. Resource Type group  10.2.7.2 Create <*group*>10.2.7.4 Update <*group*> |
| 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) | 7.4.13 in TS-0004 |
| 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

The *memberType* attribute is closely related to *consistencyStrategy* attribute. The descriptions are:

This attribute determines how to deal with the *<group>* resource if the *memberType* validation fails. Its possible values are

* ABANDON\_MEMBER
* ABANDON\_GROUP
* SET\_MIXED

Which means delete the inconsistent member if the attribute is ABANDON\_MEMBER; delete the group if the attribute is ABANDON\_GROUP; set the *memberType* to "mixed" if the attribute is SET\_MIXED.

If it is not given by the Originator at the creation procedure, default is " ABANDON\_MEMBER "

[case 1] single *memberType* of all reachable *memberIDs*

No issues.

[case 2] single memberType of partial reachable memberIDs

To deal with an unreachable *memberIDs,* it should check *consistencyStrategy.*

First, the consistencyStrategy shall ABANDON\_GROUP or ABANDON\_MEMBER in case of single memberType.

If the *consistencyStrategy* is SET\_MIXED, it is not clear to process.

Second, it is not clear to process for unreachable *memberIDs.* It depends on implementation.

[case3] mixed *memberType* and *consistencyStrategy* is SET\_MIXED

Third problem, it is not clear to process for unreachable memberIDs

This contribution proposes to resolve above issues by introducing *unreachableMemberIDs* and *enforcement* attributes.

An *unreachableMemberIDs* attribute is a list of *memberIDs* that is not reachable.

This is useful to check the validation of <group> resource. The TS-0004 is specified when *memberIDs* is not reachable, whole validation is performed again.

With this *unreachableMemberIDs* attribute, validation for <group> resource can perform to partial *memberIDs* not whole *memberIDs*

An *enforcement* attribute is value whether the unreachable *memberIDs* include or not. When enforcement is TRUE, operation(Create, Retrieve, Update, Delete) shall be performed even if *unreachableMemberIDs* exist.

When enforcement is FALSE, it may check reachability for *unreachableMemberIDs* or check *memberTypeValidated*.

The TS-0004 is not explicitly specified on this at this time.

The followings are possible procedure for protocol aspects. It needs to be discussed at PRO WG.

 

The SET\_MIXED can interpret two aspects.

First, SET\_MIXED means mixed resource type. For example, member types are AE, remoteCSE, ans son on.

Second, SET\_MIXED means mixed *memberIDs* both reachable and unreachable.

This contribution proposes for the SET\_MIXED on reachability which is not clear and describe related procedures.

Therefore, this revised contribution add change request 2 and 3 which is about procedure for <group> management based on comments during first discussion and off-line discussion.

R02:

* Rollback description of *memberType*
* Rollback description of *consistencyStrategy* with a bit of revision.

Revision 3 includes:

- e-mail discussion in oneM2M Architecture mailing list in April 2007.

- the results of the discussion ARC-2017-0167R01 which has been discussed at conference call 28.2

A summary of the e-mail discussion is as following

ARC-2016-0484R02 is agreed to use Group Request Target Members parameter.

How does Originator knows or maintains fanout for Group Request Target Members ?   
In TS-0004, fanoutpoint CRUD operations, for the Originator, we have mentioned (in PRO CR:  [PRO-2016-0378R04](http://member.onem2m.org/Application/documentApp/documentinfo/?documentId=20363&fromList=Y" \t "_blank)) that after receiving the response of fanoutpoint operation, the Originator upon analyzing the individual responses in an aggregated response knows that operation failed for which members. Now the Originator may issue another request with Group Request Target Members parameter set for failed members.

How to handle unreachable memberIDs at step of validity check at a Receiver for CREATE operation ?   
For <group> CREATE request, currently it is the internal handling of CSE to keep the mapping of such unreachable members.

A summary of the ARC-2017-0167R01 is as following

It presents a rationale for ARC-2017-0115 and introduces <*uFanOutPoint*> as a virtual resource which aims to handle (e.g. wake up or trigger to ureachableMemberIDs)

The *<uFanOutPoint>* resource is a virtual resource because it does not have a representation. It is the child resource of a *<group>* resource. Whenever a request is sent to the *<uFanOutPoint>* resource, the request is fanned out to each of the members of the *<group>* resource indicated by the *unreachableMemberIDs* attribute of the *<group>* resource. Before an execution of <*uFanOutPoint*>, triggering action may required. The responses (to the request) from each member are returned to the Originator. A timer should be set for the aggregation.

The *<uFanOutPoint>* resource does not have a resource representation by itself and consequently it does not have an *accessControlPolicyIDs* attribute. The *<accessControlPolicy>* resource used for access control policy validation is indicated by the *membersAccessControlPolicyIDs* attribute in the parent *<group>* resource.

Issues from conference call 28.2

A list of *unreachableMemberIDs* maychange from time to time*.* How to maintain *unreachableMemberIDs* at Hosting CSE ?

Possible solution for proposal

A memberIDs for <group> resource is a list of member resource IDs which exist under CSEBase, AE or remoteCSE.

The <remoteCSE> and <AE> resources have specified *requestReachability* attribute in clause 9.6.4 and 9.6.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Attributes* | Multiplicity | RW/  RO/  WO | Description | Annc Attributes |
| *requestReachability* | 1 | RW | If the CSE that created this *<remoteCSE>* resource can receive a request from other AE/CSE(s), this attribute is set to "TRUE" otherwise "FALSE" (see note) | OA |

Therefore any resource for *memberIDs* can provide the reachability information of its <remoteCSE> or <AE> resource form the stem resource to Hosting CSE of <group> resource.

A Hosting CSE of <group> resource get reachability information by subscription i.e. the Hostong CSE of <group> resource subscribes <remoteCSE> or <AE> resource of each *memberIDs*.

The *notificationURI* attribute of the <subscription> resource is <group> resource.

Hosting CSE of <group> resource extracts reachability information form the notification of subscription result which will be updated information of reachability. ----------------------Start of change 1-------------------------------------------

### 9.6.13 Resource Type *group*

The *<group>* resource represents a group of resources of the same or mixed types. The *<group>* resource can be used to do bulk manipulations on the resources represented by the *memberIDs* attribute. The *<group>* resource contains an attribute that represents the members of the group and the *<fanOutPoint>* virtual resource that enables generic operations to be applied to all the resources represented by those members. By grouping <*semanticDescriptor*> resources across which a semantic description is distributed, another virtual resource (<*semanticFanOutPoint*>) enables semantic discovery procedures to be applied across the full logical tree in the description.



Figure 9.6.13-1: Structure of *<group>* resource

The *<group>* resource shall contain the child resources specified in table 9.6.13-1.

Table 9.6.13-1: Child resources of <group> resource

| Child Resources of *<group>* | Child Resource Type | Multiplicity | Description | *<groupAnnc>* Child Resource Types |
| --- | --- | --- | --- | --- |
| *[variable]* | *<semanticDescriptor>* | *0..n* | *See clause 9.6.30* | *<semanticDescriptor>, <semanticDescriptorAnnc>* |
| *[variable]* | *<subscription>* | 0..n | See clause 9.6.8 | *<subscription>* |
| *fopt* | *<fanOutPoint>* | 1 | See clause 9.6.14 | none |
| *sfop* | *<semanticFanOutPoint>* | 0..1 | See clause 9.6.14a | none |

The *<group>* resource shall contain the attributes specified in table 9.6.13-2.

Table 9.6.13-2: Attributes of *<group>* resource

| Attributes of  *<group>* | Multiplicity | RW/  RO/  WO | Description | *<groupAnnc>* 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 |
| *accessControlPolicyIDs* | 0..1 (L) | 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 |
| *dynamicAuthorizationConsultationIDs* | 0..1 (L) | RW | See clause 9.6.1.3. | OA |
| *creator* | 0..1 | RO | See clause 9.6.1.3. | NA |
| *memberType* | 1 | WO | It is the resource type of the member resources of the group, if all member resources (including the member resources in any sub-groups) are of the same type. Otherwise, it is of type 'mixed'. | OA |
| *currentNrOfMembers* | 1 | RO | Current number of members in a group. It shall not be larger than *maxNrOfMembers*. | OA |
| *maxNrOfMembers* | 1 | RW | Maximum number of members in the *<group>*. | OA |
| *memberIDs* | 1 (L) | RW | List of member resource IDs referred to in the remaining of the present document as *memberID*. Each ID (*memberID*) should refer to a member resource or a (sub-) *<group>* resource of the *<group>*. A <group> resource with an empty member list is allowed. | OA |
| *membersAccessControlPolicyIDs* | 0..1 (L) | RW | List of IDs of the *<accessControlPolicy>* resources defining who is allowed to access the *<fanOutPoint>* resource. | OA |
| *memberTypeValidated* | 0..1 | RO | Denotes if the resource types of all members resources of the group has been validated by the Hosting CSE. In the case that the *memberType* attribute of the <*group*> resource is not 'mixed', then this attribute shall be set.. | OA |
| *consistencyStrategy* | 1 | WO | This attribute determines how to deal with the *<group>* resource if the *memberType* validation fails. Its possible values are   * ABANDON\_MEMBER * ABANDON\_GROUP * SET\_MIXED   Which means delete the inconsistent member if the attribute is ABANDON\_MEMBER;  delete or avoid execution of the group if the attribute is ABANDON\_GROUP;  set the *memberType* to "mixed" if the attribute is SET\_MIXED  If it is not given by the Originator at the creation procedure, default is " ABANDON\_MEMBER " | OA |
| *groupName* | 0..1 | RW | Human readable name of the *<group>*. | OA |
| *semanticSupportIndicator* | 0..1 | RO | This is boolean value. When it is TRUE, <*group*> shall be support for sematic discovery functionality via <semanticFanOutPoint>. | OA |
| *unreachableMemberIDs* | 0..1(L) | RW | List of *memberIDs* that is not reachable.This is dynamically updated by subscripon | NA |
| *enforcement* | 1 | WO | This is boolean value and default is TRUE. When *enforcement* is TRUE, operation shall be performed even if *unreachableMemberIDs* exist.  When *enforcement* is FALSE, it may check reachability for *unreachableMemberIDs* or check *memberTypeValidated.* | MA |



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

-----------------------Start of change 2---------------------------------------------

#### 10.2.7.2 Create *<group>*

This procedure shall be used for creating a *<group>* resource.

Table 10.2.7.2-1: *<group>* CREATE

|  |  |
| --- | --- |
| *<group>* CREATE | |
| Associated Reference Point | Mcc, Mca and Mcc' |
| Information in Request message | ***From:*** Identifier of the AE or the CSE that initiates the Request  ***To:*** The address of the *<CSEBase>, <AE>, or <remoteCSE>* where the *<group>* resource is intended to be Created  ***Content:*** The representation of the *<group>* resource for which the attributes are described in clause 9.6.13 |
| Processing at Originator before sending Request | The Originator shall request to Create a *<group>* resource by using the CREATE operation. The request shall address *<CSEBase>, <remoteCSE> or <AE>* resource of a Hosting CSE. The Request shall also provide *memberIDs* and may provide *expirationTime* attributes. The Originator may be an AE or a CSE |
| Processing at Receiver | For the CREATE procedure, the Receiver shall:   * Check if the Originator has CREATE permissions on the target resource * Check the mandatory parameters, mandatory attributes (e.g. universal attributes, common attributes and <group> specific attributes) and optional attributes. * Validate that there are no duplicate members present in the *memberIDs* attribute * Validate that the resource type of every *memberIDs* of the *memberType* attribute in the request, if the *memberType* attribute of the *<group>* resource is not 'mixed'. Set the *memberType* attribute to single resource type of *memberIDs*. * Upon successful validation of the provided attributes, create a new group resource including the *<fanOutPoint>* child-resource in the Hosting CSE. If the CSE supports semantic discovery functionality, the Hosting CSE shall also create and set the *semanticSupportIndicator* attribute to TRUE and create the <*semanticFanOutPoint>* child-resource. * In the case that the group resource contains unreachable *memberIDs* or sub-group resources as member resource, set the *memberTypeValidated* attribute of the *<group>* resource to FALSE * Respond to the Originator with the appropriate Response with the representation of the *<group>* resource based on *consistencyStrategy* if the *memberTypeValidated* attribute is FALSE. * As soon as any unreachable resource becomes reachable, the *unreachableMemberIDs* shall be updated. If the *memberIDs* still unreachable, the Hosting CSE shall deal with the *<group>* resource according to the policy defined by the *consistencyStrategy* attribute of the *<group>* resource provided in the request. or by default if the attribute is not provided |
| Information in Response message | The representation of the *<group>* resource if the *memberTypeValidated* attribute is FALSE |
| Processing at Originator after receiving Response | None |
| Exceptions | No change from the basic procedure in clause 10.1.2 |

-----------------------End of change 2---------------------------------------------

-----------------------Start of change 3---------------------------------------------

#### 10.2.7.4 Update *<group>*

This procedure shall be used for updating an existing *<group>* resource.

Table 10.2.7.4-1: *<group>* UPDATE

|  |  |
| --- | --- |
| *<group>* UPDATE | |
| Associated Reference Point | Mca, Mcc and Mcc' |
| Information in Request message | ***From:*** Identifier of the AE or the CSE that initiates the Request  ***To:*** The address of the *<group>* resource |
| Processing at Originator before sending Request | The Originator shall request to update attributes of an existing *<group>* resource by using an UPDATE operation. The Request shall address the specific *<group>* resource of a CSE. The Originator may be an AE or a CSE |
| Processing at Receiver | The UPDATE procedure shall be:   * Check if the Originator has UPDATE permissions on the *<group>* resource. * Check mandatory parameters, mandatory attributes (e.g. universal attributes, common attributes and <group> specific attributes) and optional attributes. * Validate that there are no duplicated members present in the *memberIDs* attribute * Validate that the resource type of every *memberIDs* of the *memberType* attribute in the request, if the *memberType* attribute of the *<group>* resource is not 'mixed'. Set the *memberType* attribute to single resource type of *memberIDs*. * Upon successful validation of the provided attributes, update the *<group>* resource in the Hosting CSE * In the case that the *<group>* resource contains unreachable *memberIDs* or sub-group resources as members resource set the *memberTypeValidated* attribute of the *<group>* resource to FALSE * Respond to the Originator with the appropriate response with the representation of the *<group>* resource based on *consistencyStrategy* if the *memberTypeValidated* attribute is FALSE. * As soon as any unreachable resource becomes reachable, the *unreachablememberIDs* shall be updated. If the *memberIDs* still unreachable, the Hosting CSE shall deal with the *<group>* resource according to the policy defined by the *consistencyStrategy* attribute of the *<group>* resource provided in the request, or by default if the attribute is not provided |
| Information in Response message | The representation of the <group> resource if the *memberTypeValidated* attribute is FALSE |
| Processing at Originator after receiving Response | None |
| Exceptions | No change from the basic procedure in clause 10.1.4 |

-----------------------End of change 3---------------------------------------------

-----------------------Start of change 4---------------------------------------------

### 9.6.14.2 Resource Type *uFanOutPoint*

The *<uFanOutPoint>* resource is a virtual resource because it does not have a representation. It is the child resource of a *<group>* resource. Whenever a request is sent to the *<uFanOutPoint>* resource, the request is fanned out to each of the members of the *<group>* resource indicated by the *unreachableMemberIDs* attribute of the *<group>* resource. Before an execution of <*uFanOutPoint*>, triggering action may required. The responses (to the request) from each member are returned to the Originator. A timer should be set for the aggregation.

The *<uFanOutPoint>* resource does not have a resource representation by itself and consequently it does not have an *accessControlPolicyIDs* attribute. The *<accessControlPolicy>* resource used for access control policy validation is indicated by the *membersAccessControlPolicyIDs* attribute in the parent *<group>* resource.

----------------------End of change 4---------------------------------------------

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 include a proposal to change only 3 tables?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 clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the new clause 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.?