|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting ID:\* | PRO 31 |
| Source:\* | Poornima, C-DOT, poornima@cdot.inSuman, C-DOT, ssheoran@cdot.inYongjing, Huawei, zhangyongjing@huawei.com  |
| Date:\* | 2017-08-29 |
| Reason for Change/s:\* | See the introduction  |
| CR against: Release\* | Release 2 |
| CR against: WI\* | [ ]  Active <Work Item number> [x]  MNT maintenance / < Work Item number(optional)>Is this a mirror CR? Yes [x]  No mirror CR number: ARC-2017-0355(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-0001 Version 2.15.0 |
| Clauses \* | 9.6.13, 10.2.7.2, 10.2.7.4 |
| 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 |
| Impacted other TS/TR(s) | <TS/TR number>, <Version Number>, and <Description on which aspect should be reflected in this TS/TR> |
| 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 [ ]  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, memberIDs denotes list of member resource identifiers, these ids refers to member resource or sub-group resource as highlighted below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *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 |

Problem 1: how do we know that which member is of type sub-group as mentioned above?

Problem 2: it is not clear when a member of type <group> is to be treated as member or sub-group resource. Currently all the members of type group will be treated as sub-group as per handling given in TS-0004.

**TS-0004**

|  |
| --- |
| The primitive parameters ***From*** and ***To*** shall be mapped to the primitive parameters of the corresponding Request to be sent out to each member of the group. The primitive parameter ***From*** shall be directly used. The primitive parameter ***To*** (i.e. <URI of group resource>/fanOutPoint) shall be replaced by resource identifiers present in the ***memberIDs*** attribute of the group resource, but excluding the member resources which are sub-groups. For member resources that are sub-groups, the ***To*** parameter of the primitives shall be the resource identifier of the *fanOutPoint* virtual resource of the sub-group resources. In addition, any additional relative address that was appended to .../fanOutPoint in the original Request shall be appended to each ***To*** URI. For those members resource contained in a sub-group, the primitive ***To*** of the composed Request shall be <URI of sub-group resource>/fanOutPoint plus any additional appended relative address including in the original Request. The group hosting CSE shall execute "Compose Request primitives". In addition, the group hosting CSE shall generate a unique group request identifier, add it as a primitive parameter to the Request and locally store the group request identifier as per the local policy. |

**Examples:**



Figure 1 : <group1> resource, members are of type containers residing on CSE1 and of type group residing on CSE2.

In figure1, First, to distinguish group type of members, CSE needs to retrieve the resource type information.

Secondly, when there is a member of type group, then it is not clear that whether a fanout request is to be sent or a normal request is to be sent as cse1 doesn’t know that it’s a member resource or sub-group resource

The issue was discussed in the forum and one solution is to validate the each memberType for both the cases mixed or same type of member and keep the memberType information in <group> resource attribute or internally in CSE. Another way could be to put the onus on originator by passing “/fopt” in group type of memberIDs when originator wants to treat that member as sub-group.

The CR proposes to distinguish sub-group members by appending “/fopt” in memberIDs for group type members as discussed in the forum.

The CR proposes to add description in **memberIDs** attribute to resolve the issue. Also details added in Originator processing to resolve the concern.

### -----------------------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> if memberID is suffixed with “/fopt”*. 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 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 | Indicator of support for sematic discovery functionality via <semanticFanOutPoint>. | OA |

### -----------------------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. For members which are of type <group>, the originator shall suffix the ‘/fopt’ to that ‘memberID‘ during group creation if the originator wants to fan-out the group request to each member of that sub-<group>, else originator shall not suffix the ‘/fopt’ to that ‘memberID‘. 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 validity of the provided attributes
* Validate that there are no duplicate members present in the *memberIDs* attribute
* Validate that the resource type of every member on each member Hosting CSE conforms to the *memberType* attribute in the request, if the *memberType* attribute of the *<group>* resource is not 'mixed'. Set the *memberTypeValidated* attribute to TRUE upon successful validation
* 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
* Conditionally, in the case that the group resource contains temporarily. unreachable Hosting CSE of sub-group resources as member resource, set the *memberTypeValidated* attribute of the *<group>* resource to FALSE
* Respond to the Originator with the appropriate generic Response with the representation of the *<group>* resource if the *memberTypeValidated* attribute is FALSE, and the address of the created *<group>* resource if the CREATE was successful
* As soon as any Hosting CSE that hosts the unreachable resource becomes reachable, the *memberType* validation procedure shall be performed. If the *memberType* validation fails, 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 10.1.1 |

### -----------------------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. If originator intends to update memberIDs attribute ,for members which are of type <group>, originator shall suffix the ‘/fopt’ to that ‘memberID‘ during group updation if the originator wants to fan-out the group request to each member of that sub-<group> ,else originator shall not suffix the ‘/fopt’ to that ‘memberID‘. 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 the validity of provided attributes
* Validate that there are no duplicated members present in the *memberIDs* attribute
* Validate that the resource type of every member on each member Hosting CSE conforms to the *memberType* attribute in the request, if the *memberType* attribute of the *<group>* resource is not 'mixed'. Set the *memberTypeValidated* attribute to TRUE upon successful validation
* Upon successful validation of the provided attributes, update the *<group>* resource in the Hosting CSE
* Conditionally, in the case that the *<group>* resource contains temporarily unreachable Hosting CSE of sub-group resources as members resource set the *memberTypeValidated* attribute of the *<group>* resource to FALSE
* Respond to the Originator with the appropriate generic response with the representation of the *<group>* resource if the *memberTypeValidated* attribute is FALSE, and the address of the created *<group>* resource if the UPDATE is successful
* As soon as any Hosting CSE that hosts unreachable resource becomes reachable, the *memberType* validation procedure shall be performed. If the *memberType* validation fails, 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 10.1.3 |

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

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.?