|  |
| --- |
| Input Contribution |
| Meeting ID\* | ARC#29 |
| Title:\* | Potential solution for cross-resource subscription |
| Source:\* | SeungMyeong JEONG, KETI sm.jeong@keti.re.krSungchan Choi, KETI, csc@keti.re.kr |
| Date:\* | 2017-05-18 |
| Input related to\* | TR-0026 |
| Intended purpose ofdocument:\* | [ ]  Decision[x]  Discussion[ ]  Information[ ]  Other <specify> |
| Impacted other TS/TR(s) | n/a |
| Decision requested or recommendation:\* | Incorporate the text into TR-0026 |
| 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

This contribution proposes a new possible solution for cross-resource subscription.

### -----------------------Start of input 1-------------------------------------------

## 10.X Solution C: Cross-Resource Subscription #2

### 10.X.1 Solution Description

In order to address the Key Issue 5, a subscriber AE/CSE creates resource subscriptions where automatic notifications depend on two or more resources, not a single resource as described in the previous solution. In this clause, a new candidate solution for the same use case but with different resource types and procedures is illustrated.

In short, a new resource type *subscriptionAssociation* is suggested which has two-ways linkes from/to exsiting *subscription* resource type. When a <subscriptionAssociation> resource is created which associates exsiting <subscription> resources, notifications generated per <subscription> resource is sent to the <subscriptionAssociation> resource Hosting CSE to generate a single but not just aggregated notification to the cross-resource subscriber.

The main difference between the previous and this one is whether the solution relies on group functionality (i.e. *group* resource type) or not while making cross-resource subscription relationship.

### 10.X.2 Solution Applicability

This solution applies to Key Issue 5.

### 10.X.3 New Resources and Procedures

#### 10.X.3.1 Introduction

A new resource <*subscriptionAssociation*> is proposed to implement Cross-Resource Subscription functionality, associating with existing <*subscription*> resources.

Basically two-way links are maintained between a <*subscriptionAssociation*> resource and <*subscription*> resource(s). So that individual notification per <*subscription*> resource is sent to the <*subscriptionAssociation*> resource Hosting CSE referring the link in the <*subscription*> resource.

This link is also used to manage the list of individual subscriptions in the <*subscriptionAssociation*> resource. E.g. when a <*subscription*> resource is removed, then the link to the resource in the <*subscriptionAssociation*> resource is removed also. If there is a newly created resource subscription and needs to be associated with the exsiting <*subscriptionAssociation*> resource, then the subscriber can just add the link to that <*subscription*> resource. There’s no need to do something for the <*group*> resource update.

To have a single event notification from cross or multiple resource subscriptions, the *timeWindow* attribute is suggested to check all the individual events are occurred during the time window, so the single notification needs to be sent to the subscriber of the <*subscriptionAssociation*> resource.

#### 10.X.3.2 New *subscriptionAssociation* resource type

The new resource type *subscriptionAssociation* is suggested to have link/association to existing <*subscription*> resources. The Hosting CSE of this resource receives notifications from the <*subscription*> resource Hosting CSEs, and generate a single notification to the subscriber when all the events from individual subscription occurred in a time window that is set by the subscriber. The subscriber considers those events occurred within the window consist of a meaningful information so the single notification for that is enough.

Table 10.X.3.2-1: Child resources of *<subscriptionAssociation>* resource

| Child Resources of *<container>* | Child Resource Type | Multiplicity | Description |
| --- | --- | --- | --- |
| *subDel* | *<subscriptionLinkDeletion>* | 1 | This is the virtual resource only permits DELETE operation. The <subscription> Hosting CSE in the *subsciptionIDs* list is allowed to delete the <subscription> resource from the list. |

**Table 10.X.3.2-2: Resource specific attributes of *<subscriptionAssociation>* resource**

| Attributes of *<subscriptionAssociation>* | Multiplicity | RW/RO/WO | Description |
| --- | --- | --- | --- |
| *subscriptionIDs* | 1 (L) | RW | This attribute indicates the resource address(es) of associated <*subscription*> resources. |
| *timeWindow* | 1 | RW | This attribute indicates the time duration (e.g. in seconds) that the cross-resource subscriber considers all associated subscription events are occurred in the time window so that a single notification is need for those event. |
| *notificationType* | 1 | RW | This indicates the type of information for notifications of this <*subscriptionAssociation*> resource. Possible values are 1) simple notice that all associated subscriptions have events in the time window, and 2) aggregation of the notifications from associated subscriptions. |

#### 10.X.3.3 Extension to *subscription* resource type

A new attribute is added to have a link to <*subscriptionAssociation*> resource. This is used to remove the link from the associated subscription to the individual subscription as well as to send a notification to the <*subscriptionAssociation*> resource Hosting CSE.

**Table 10.X.3.3-1: New attributes of *<subscription>* resource**

| Attributes of *<subscription>* | Multiplicity | RW/RO/WO | Description |
| --- | --- | --- | --- |
| *associatedSub* | 0..n | RW | This attribute lists *eventNotificationCriteria* for each target resource involved in a cross-resource subscription.  |

#### 10.X.3.4 Procedure to create subscription association

When an Originator requests to create a <*subscriptionAssociation*> resource, after the basic checking procedures for resource creation as defined in TS-0001, the Hosting CSE creates the <*subscriptionAssociation*> resource. After this, the Hosting CSE sends UPDATE request(s) to <*subscription*> resources indicated in the *subscriptionIDs* attribute to include the *associatedSub* attribute targeting the created <*subscriptionAssociation*> resource..

#### 10.X.3.5 Procedure to manage subscription association

After the creation of a <*subscriptionAssociation*> resource, when the subscriber wants to associate a new <*subscription*> resource to the existing subscription association, the <subscription> Hosting CSE sends UPDATE Request to the <*subscriptionAssociation*> resource to update the *subscriptionIDs* attribute.

After the creation of a <*subscriptionAssociation*> resource, when an associated <*subscription*> resource having the associatedSub attribute gets deleted, the <*subscription*> Hosting CSE sends DELETE Request to the virtual child resource, which has the fixed name “subDel”, of the <*subscriptionAssociation*> resource. Then the <*subscriptionAssociation*> resource does not wait for notifications coming from the <*subscription*> Hosting CSE anymore to generated a single event to the subscriber.

#### 10.X.3.6 Procedure to generate notifications of cross-resource subscription

After the creation of a <*subscriptionAssociation*> resource, when the subscriber wants to associate a new <*subscription*> resource to the existing subscription association, the <subscription> Hosting CSE sends UPDATE Request to the <*subscriptionAssociation*> resource to update the *subscriptionIDs* attribute.

### -----------------------End of input 1---------------------------------------------