|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting ID:\* | SDS 46.2 |
| Source:\* | Bob Flynn, Convida Wireless , Bob.Flynn@convidawireless.comAndreas Kraft, Deutsche Telekom, Andreas.Kraft@t-systems.com Miguel Angel Reina Ortega, ETSI, MiguelAngel.ReinaOrtega@etsi.org  |
| Date:\* | 2020-08-26 |
| Reason for Change/s:\* | Announcement Procedure - Update |
| CR against: Release\* | Rel-4 |
| CR against: WI\* | [x]  Active < WI-0077> [ ]  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-0001 v4.7.0 |
| Clauses \* | 9.6.3, 10.2.13.2, 9.6.26.1 |
| Type of change: \* | [ ]  Editorial change[ ]  Bug Fix or Correction[x]  Change to existing feature or functionality[ ]  New feature or functionalityOnly ONE of the above shall be ticked |
| Other TS/TR(s) impacted | None |
| 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 2019 (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

While working on announcement development / test cases, there were some issues that we wanted to highlight for discussion and propose a solution for those issues.

R01 – Some changes in the resource structure and message flow introduction.

R02 – Some more minor changes to the resource structure and message flow introduction

R03 – Allow normal resources as child of CSEBaseAnnc

R04 – Added a new change on table 9.6.1.1-1 for CSEBaseAnnc as parent of certain resources

Consider the scenario where <aResource> is to be announced to CSE1, CSE2, and CSE5.



First thing to point out is that the announced resource is in different locations depending on the “registration” status/relationship to Hosting CSE.

The second thing to point out is that when CSE1 deregisters from Hosting CSE then all of the announced resources are DELETED. Whereas if CSE2 de-registers from CSE1, the announced resources remain intact (for example until re-registration). This is a consequence of the first issue above.

The third thing to point out is the specification wording of the announce procedure. Specifically, from 10.2.13.2-1:

The original resource Hosting CSE shall first check if it is a Registree or the Registrar of the announcement target CSE. If that is the case, the announced resource shall be created as a direct child of the Hosting CSE’s <*remoteCSE*> resource hosted by the announcement target CSE. If that is not the case, the Hosting CSE shall next check if its <*remoteCSE*> resource has been announced to the announcement target CSE. The Hosting CSE shall perform this check by checking the *announceTo* attribute of its <*remoteCSE*> resource hosted on its Registrar CSE if the announcement target CSE is not a descendent CSE, or the corresponding Registree CSE if the announcement target CSE is a descendent CSE. If it is not announced, **the Hosting CSE shall request that its Registrar CSE (If the target CSE is not its descendant CSE) or Registree CSE (if the target CSE is its descendant CSE) to create a <*remoteCSEAnnc*> resource representing the Hosting CSE as a direct child of the <*CSEBase*> of the announcement target CSE.** The announced resource shall then be created by the Hosting CSE as a direct child resource of the <*remoteCSEAnnc*> resource.

This requires the Hosting CSE to update the ‘announceTo’ attribute of its <remoteCSE> to include the announcement target CSE. At this point, the proper execution become unclear:

It is intended that CSE1 would send a CREATE <remoteCSEAnnc> to the announcement target CSE.

Because the announced resource is a <remoteCSE> CSE1 is NOT supposed to follow the FULL procedure described above, i.e. check if CSE1 is registered to the announcement target CSE. This is not described clearly (perhaps a TS-0004 level of detail).

A proposed solution to simplify the procedure, eliminate the inconsistent parent resource type and address the deleted announced resources during de-registration is:

* to make <cseBase> announceable
* Then, to remove the registration check to determine the target of the announcement as target of the announcement will always be a cseBaseAnnc
* remove the decendent cse check because normal retargeting rules will apply.





The benefits of these proposed changes are:

There is a consistent location for announced resources (always under <CSEBaseAnnc>)

The newly defined announce procedure applies with no exceptions.

The original resource hosting CSE is able to self manage the target of the announced resources (CSEBaseAnnc)

Separates registration and announcement concerns

 Now announced resource will remain after deregistration of a CSE

## ----------------------- Start of Change 1--------------------------------------------

#### 10.2.13.2 Procedure for AE and CSE to initiate Creation of an Announced Resource

This clause describes the procedure for an AE or a CSE to initiate the creation of an announced resource.

Figure 10.2.13.2-1 depicts how creation of an announced resource is initiated (clause 10.2.13.2) and the announced resource is created on an announcement target CSE (clause 10.2.13.5).



Figure 10.2.13.2-1: Announced resource CREATE procedures

The Originator of a Request for initiating resource announcement can be either an AE or a CSE. Two methods are supported for initiating the creation of an announced resource:

* CREATE: The Originator can initiate the creation of an announced resource during the creation of the original resource by providing *announceTo* attribute in the CREATE Request.
* UPDATE: The Originator can initiate the creation of an announced resource by using the UPDATE Request to update the *announceTo* attribute at the original resource.

Table 10.2.13.2-1: Initiate Resource Announcement: UPDATE or CREATE

| *Initiate Resource Announcement:* CREATE or UPDATE |
| --- |
| Information in Request message | All parameters defined in table 8.1.2-3 are applicable as indicated in that table. In addition, for the case of the CREATE procedure for a specific resource is described in clause 10.2. The Originator suggests the address(es) or the CSE-ID(s) to which the resource will be announced in the ***Content*** parameter. |
| Processing at the Originator before sending Request  | ***Content*:** contains address where the resource needs to be announced (within *announceTo* attribute):* The Originator provides either the address(es) for the announced resource or the list of CSE-IDs of the remote CSEs where the original resource needs to be announced by including such information within the *announceTo* attribute of the UPDATE or CREATE Request.
 |
| Processing at the Receiver | Once the Originator has been successfully authorized, the Receiver (which shall be the original resource Hosting CSE) shall grant the Request after successful validation of the Request:* If the Request provides address(es) for the announced resource that are not already stored in the *announceTo* attribute or for newly created *announceTo* attribute, the Receiver shall announce the resource to the announcement target CSE.
* If the Request provides a list of CSE-IDs of the remote CSEs that are not already stored in the *announceTo* attribute of for the newly created or updated *announceTo* attribute, the Receiver shall decide the location at the remote CSE(s) identified by CSE-ID(s) and announce the resource to the announcement target CSE.

The original resource Hosting CSE shall first check if <CSEBase> is announced to the announcement target CSE by checking the *announceTo* attribute of <CSEBase>. If that is not the case, the Hosting CSE shall send a CREATE <CSEBaseAnnc> to the announcement target CSE. The announced resource shall then be created by the Hosting CSE as a direct child resource of the <CSEBaseAnnc> resource.  |
| Information in Response message | On successful completion of resource announcement as in clause 10.2.3.5, the Receiver shall provide all parameters defined in table 8.1.3-1 that are applicable as indicated in that table in the Response message:* The Receiver shall provide the address(es) of the announced resource to the Originator by updating the content of the *announceTo* attribute in the original resource and by providing it in the UPDATE or CREATE Response message depending on the type of the Request.
 |
| Processing at Originator after receiving Response | According to clause 10.1.2 in case of CREATE Request.According to clause 10.1.4 in case of UPDATE Request. |
| Exceptions | All exceptions described in the basic procedures (clause 10.1.2) are applicable. |

-------------------------------------------------- End of Change 1---------------------------------------------------

## ----------------------- Start of Change 2--------------------------------------------

### 9.6.3 Resource Type *CSEBase*

A *<CSEBase>* resource shall represent a CSE. The *<CSEBase>* resource shall be the root for all resources that are residing in the CSE. A CSE shall be represented by only one *<CSEBase>* resource.

The *<CSEBase>* resource shall contain the child resources specified in table 9.6.3-1.

Table 9.6.3-1: Child resources of *<CSEBase>* resource

|  |  |  |  |
| --- | --- | --- | --- |
| Child Resources of *<CSEBase>* | Child Resource Type | Multiplicity | Description |
| *[variable]* | *<cseBaseAnnc>* | 0..n | Announced variant of <cseBase>. Resource with CSE-specific information for a CSE that intends to announce resources to another CSE |
| *[variable]* | *<remoteCSE>* | 0..n | See clause 9.6.4 |
| *[variable]* | *<remoteCSEAnnc>* | 0..n | Announced variant of <*remoteCSE>*. Resourcewith CSE-specific information for a CSE that announced itself to another CSE with which it does not have a registration relationship. |
| *[variable]* | *<node>* | 0..n | See clause 9.6.18 |
| *[variable]* | *<AE>* | 0..n | See clause 9.6.5 |
| *[variable]* | *<container>* | 0..n | See clause 9.6.6 |
| *[variable]* | *<flexContainer>* | 0..n | See clause 9.6.35 |
| *[variable]* | *<group>* | 0..n | See clause 9.6.13 |
| *[variable]* | *<accessControlPolicy>* | 0..n | See clause 9.6.2 |
| *[variable]* | *<subscription>* | 0..n | See clause 9.6.8 |
| *[variable]* | *<mgmtCmd>* | 0..n | See clause 9.6.16 |
| *[variable]* | *<locationPolicy>* | 0..n | See clause 9.6.10 |
| *[variable]* | *<statsConfig>* | 0..n | See clause 9.6.23 |
| *[variable]* | *<statsCollect>* | 0..n | See clause 9.6.25 |
| *[variable]* | *<request>* | 0..n | See clause 9.6.12 |
| *[variable]* | *<delivery>* | 0..n | See clause 9.6.11 |
| *[variable]* | *<schedule>* | 0..1 | This resource defines the reachability schedule information of the entity. The absence of this resource implies the entity is always reachable. See clause 9.6.9 |
| *[variable]* | *<role>* | 0..n | See clause 9.6.38 |
| *[variable]* | *<token>* | 0..n | See clause 9.6.39 |
| *[variable]* | *<m2mServiceSubscriptionProfile>* | 0..n | See clause 9.6.19 |
| *[variable]* | *<serviceSubscribedAppRule>* | 0..n | See clause 9.6.29 |
| *[variable]* | *<notificationTargetPolicy>* | 0..n | See clause 9.6.32 |
| *[variable]* | *<dynamicAuthorizationConsultation>* | 0..n | See clause 9.6.40 |
| *[variable]* | *<timeSeries>* | 0..n | See clause 9.6.36 |
| *[variable]* | *<authorizationDecision>* | 0..n | See clause 9.6.41 |
| *[variable]* | *<authorizationPolicy>* | 0..n | See clause 9.6.42 |
| *[variable]* | *<authorizationInformation>* | 0..n | See clause 9.6.43 |
| *[variable]* | *<localMulticastGroup>* | 0..n | See clause 9.6.44 |
| *[variable]* | *<transactionMgmt>* | 0..n | See clause 9.6.47 |
| *[variable]* | *<transaction>* | 0..n | See clause 9.6.48 |
| *[variable]* | *<ontologyRepository>* | 0..1 | See clause 9.6.50 |
| *[variable]* | *<semanticMashupJobProfile>* | 0..n | See clause 9.6.53 |
| *[variable]* | *<semanticMashupInstance>* | 0..n | See clause 9.6.54 |
| *[variable]* | *<AEContactList>* | 0..n | See clause 9.6.45 |
| *[variable]* | *<e2eQosSession>* | 0..1 | See clause 9.6.63 |
| *[variable]* | *<nwMonitoringReq>* | 0..n | See clause 9.6.64 |
| *[variable]* | *<semanticRuleRepository>* | 0..1 | See clause 9.6.65 |
| *[variable]* | *<softwareCampaign>* | 0..n | See clause 9.6.76 |

The *<CSEBase>* resource shall contain the attributes specified in table 9.6.3-2.

Table 9.6.3-2: Attributes of *<CSEBase>* resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attributes of *<CSEBase>* | Multiplicity | RW/RO/WO | Description | *<CSEBase>* Attributes |
| *resourceType* | 1 | RO | See clause 9.6.1.3. | NA |
| *resourceID* | 1 | RO | See clause 9.6.1.3. | NA |
| *resourceName* | 1 | RO | See clause 9.6.1.3. | NA |
| *parentID* | 1 | RO | See clause 9.6.1.3. Shall be an empty string. | NA |
| *creationTime* | 1 | RO | See clause 9.6.1.3. | NA |
| *lastModifiedTime* | 1 | RO | See clause 9.6.1.3. | NA |
| *accessControlPolicyIDs* | 0..1 (L) | RO | See clause 9.6.1.3. | MA |
| *labels* | 0..1 (L) | RO | See clause 9.6.1.3. | MA |
| *announceTo* | 0..1 (L) | RO | See clause 9.6.1.3. | NA |
| *announcedAttribute* | 0..1 (L) | RO | See clause 9.6.1.3. | NA |
| *dynamicAuthorizationConsultationIDs* | 0..1 (L) | RO | See clause 9.6.1.3. | OA |
| *owner* | 0..1 | RW | See clause 9.6.1.3 | NA |
| *location* | 0..1 | RW | See clause 9.6.1.3. | OA |
| *cseType* | 0..1 | RO | Indicates the type of CSE represented by the created resource:* Mandatory for an IN-CSE, hence multiplicity (1).
* Its presence is subject to SP configuration in case of an ASN-CSE or a MN-CSE.
 | OA |
| *CSE-ID* | 1 | RO | The CSE identifier in SP-relative CSE-ID format (clause 7.2). | OA |
| *supportedResourceType* | 1 (L) | RO | List of the resource types which are supported in the CSE. This attribute contains subset of resource types listed in clause 9.2.  | NA |
| *pointOfAccess* | 1 (L) | RO | Represents the list of physical addresses to be used by remote CSEs to connect to this CSE (e.g. IP address, FQDN). This attribute is exposed to its Registree. | OA |
| *nodeLink* | 0..1 | RO | The *resource identifier* of a *<node>* resource that stores the node specific information of the node on which the CSE represented by this <*CSEBase*> resource resides.  | OA |
| *notificationCongestionPolicy* | 0..1 | RO | This attribute applies to CSEs generating subscription notifications. It specifies the rule which is applied when the storage of notifications for each subscriber (an AE or CSE) reaches the maximum storage limit for notifications for that subscriber. E.g. Delete stored notifications of lower *notificationStoragePriority* to make space for new notifications of higher *notificationStoragePriority*, or delete stored notifications of older *creationTime* to make space for new notifications when all notifications are of the same *notificationStoragePriority*. | OA |
| *contentSerialization* | 0..1 (L) | RO | The list of supported serializations of the ***Content*** primitive parameter for receiving a request from its registrants. (e.g. XML, JSON). The list shall be ordered so that the most preferred format comes first. | OA |
| *e2eSecInfo* | 0..1 | RO | See clause 9.6.1.3. | MA |
| *supportedReleaseVersions* | 0..1 (L) | RO | List of oneM2M release versions which are supported by the CSE. Starting with Release 2, this attribute is mandatory for a CSE. For CSEs compliant to older releases, this attribute is optional. For CSEs that do not include this attribute, the default release version shall be Release 1.  | MA |
| *currentTime* | 0..1 | RO | When the CSE receives a retrieve request targeting this resource or attribute, the CSE samples its current time (e.g. makes an OS call to get the system time) and respond with the value in this attribute. An Originator retrieving this attribute can use this time value to adjust and synchronize its time value to the time value of this CSE.  | OA |

-------------------------------------------------- End of Change 2---------------------------------------

## ----------------------- Start of Change 3--------------------------------------------

9.6.26.1 Overview

A resource can be announced to one or more remote CSEs to inform the remote CSEs of the existence of the original resource. An announced resource can have a limited set of attributes and a limited set of child resources from the original resource. The announced resource includes a link to the original resource hosted by the original resource‑Hosting CSE.

In case that the original resource is deleted, all announced resources for the original resource shall be deleted, except for *<AEAnnc>* resources that were created during the registration of an AE with AE-ID-Stem starting with "S", which shall not be deleted. If the announced resource is not deleted promptly (e.g. the announced resource is not reachable), the announced resource can be deleted later either by the original resource Hosting CSE or by the expiration of the announced resource itself. The original resource shall store the list of links for the announced resources for those purposes.

Synchronization between the attributes announced by the original resource and the announced resource shall be the responsibility of the original resource Hosting CSE. There shall not be any synchronization for children created at the original resource and the announced resource. The access control policy for the announced resource shall synchronize with the one from the original resource. In case that the attribute *accessControlPolicyIDs* is not present in the original resource it is the responsibility of the original resource Hosting CSE to choose the appropriate value depending on the policy for the original resource (e.g. take the parent *accessControlPolicyIDs* value).

The original resource shall have at least *announceTo* attribute present if the resource itself has been announced. If any of the Optional Announced (OA) attributes are also announced, then *announcedAttribute* attribute shall also be present. An AE or other CSE can request the original resource Hosting CSE for announcing the original resource to the list of CSE‑IDs or the address(es) listed in the *announceTo* attribute in the announcing request. An Update to the *announceTo* attribute will trigger new resource announcement(s) or the de-announcement(s) of the announced resource. After a successful announcement procedure the attribute *announceTo* contains only the list of address(es) of the announced resources.

In order to announce an attribute marked as **OA**(see clause 9.5.0), the attribute shall be included in the *announcedAttribute* attribute list at the original resource. The attributes included in the *announcedAttribute* attribute are announced to the announced resource. On successful announcement of the resource, such attributes shall be created at the announced resource; otherwise they shall not be present in the announced resource. Update to the *announcedAttribute* attribute in the original resource will trigger new attribute announcement or the de-announcement of the announced attribute(s). The announced attributes shall have the same value as the original resource, and synchronization between the value of the announced attributes at the original resource and the announced resource is the responsibility of the original resource Hosting CSE.

An announced resource may have child resources. In general, a child resource of an announced resource shall be of one of the resource types that are specified as possible child resource types for the original resource or of one of their associate announced resource types. However, for specific announced resource types, specific exceptions apply regarding which child resource types can occur. The details on which child resources are specified for each announced resource type are summarized in Table 9.6.26.1-1.

Child resources of the original resource can be announced independently as needed. In this case, the child resources at the announced resource shall be of the child resource’s associated announced type. When a child resource at the announced resource is created locally at the remote CSE, the child resource shall be of ordinary – i.e. not-announced – child resource type.

When a Hosting CSE of an original resource is initiating an announcement, it shall first check if <CSEBase> is announced to the announcement target CSE by checking the *announceTo* attribute of <CSEBase>.If it is not announced, the Hosting CSE shall send a CREATE <CSEBaseAnnc> to the announcement target CSE. The announced resource shall then be created as a child resource of the <*CSEBaseAnnc*> resource.

When a Hosting CSE of an original resource is initiating an announcement, the *From* parameter of the announce request shall contain either a SP-relative-CSE-ID of the Hosting CSE of the original resource if the announcement target CSE resides in the same SP domain or an Absolute-CSE-ID of the Hosting CSE of the original resource if the announcement target CSE resides in a different SP domain.

If an attribute is marked as **RO** and also marked as **MA** or **OA**, then only the attribute of the original resource shall be interpreted as **RO**. The corresponding attribute of the announced resource shall be always writable to the original resource hosting CSE to allow it to properly announce and de-announce the attribute and keep the announced attribute synchronized with the original one. Only the original resource Hosting CSE shall be allowed to update and delete the announced attribute which is created by the original resource Hosting CSE.

**Table 9.6.26.1-1: Announced Resource Types**

| **Announced Resource Type** | **Short Description** | **Child Resource Types** | **Clause** |
| --- | --- | --- | --- |
| *accessControlPolicyAnnc* | Announced variant of *accessControlPolicy* | *subscription* | 9.6.2 |
| *AEAnnc* | Announced variant of *AE* | *subscription,**container,**containerAnnc, flexContainer,**flexContainerAnnc,**group,**groupAnnc,**accessControlPolicy,**accessControlPolicyAnnc**semanticDescriptor,**semanticDescriptorAnnc,**timeSeries,**timeSeriesAnnc* | 9.6.5 |
| *containerAnnc* | Announced variant of *container* | *container,**containerAnnc, flexContainer,**flexContainerAnnc,**contentInstance,**contentInstanceAnnc,**subscription,**semanticDescriptor,**semanticDescriptorAnnc,**timeSeries,**timeSeriesAnnc* | 9.6.6 |
| *contentInstanceAnnc* | Announced variant of *contentInstance* |  *semanticDescriptor,**semanticDescriptorAnnc* | 9.6.7 |
| *CSEBaseAnnc* | Announced variant of CSEBase | *container,**containerAnnc, contentInstanceAnnc,* *flexContainer,**flexContainerAnnc,**group,**groupAnnc,**accessControlPolicy,**accessControlPolicyAnnc,**subscription,**scheduleAnnc,**timeSeries,**timeSeriesAnnc,**timeSeriesInstanceAnnc,**remoteCSEAnnc,**nodeAnnc,* *mgmtObjAnnc,**AEAnnc,**locationPolicyAnnc* | 9.6.3 |
| *flexContainerAnnc* | Announced variant of flexC*ontainer* | *container,**containerAnnc,**flexContainer,**flexContainerAnnc,**subscription,**semanticDescriptor,**semanticDescriptorAnnc,**timeSeries,**timeSeriesAnnc* | 9.6.35 |
| *groupAnnc* | Announced variant of *group* | *subscription,**semanticDescriptor,**semanticDescriptorAnnc* | 9.6.13 |
| *locationPolicyAnnc* | Announced variant of *locationPolicy* | None specified | 9.6.10 |
| *mgmtObjAnnc* | Announced variant of *mgmtObj* | *subscription* | 9.6.15 |
| *nodeAnnc* | Announced variant of *node* | *mgmtObjAnnc,**subscription, semanticDescriptor,**semanticDescriptorAnnc,**scheduleAnnc* | 9.6.18 |
| *remoteCSEAnnc* | Announced variant of *remoteCSE* | *container,**containerAnnc, contentInstanceAnnc,* *flexContainer,**flexContainerAnnc,**group,**groupAnnc,**accessControlPolicy,**accessControlPolicyAnnc,**subscription,**scheduleAnnc,**timeSeries,**timeSeriesAnnc,**timeSeriesInstanceAnnc,**remoteCSEAnnc,**nodeAnnc,* *mgmtObjAnnc,**AEAnnc,**locationPolicyAnnc* | 9.6.4 |
| *scheduleAnnc* | Announced variant of *schedule* | None specified | 9.6.9 |
| *semanticDescriptorAnnc* | Announced variant of *semanticDescriptor* | Subscription | 9.6.30 |
| *timeSeriesAnnc* | Announced variant of timeSeries | timeSeriesInstance,timeSeriesInstanceAnnc,subscription, semanticDescriptor,semanticDescrptorAnnc | 9.6.36 |
| *timeSeriesInstanceAnnc* | Announced variant of timeSetriesInstance | None specified | 9.6.37 |

-------------------------------------------------- End of Change 3---------------------------------------

## ----------------------- Start of Change 4--------------------------------------------

**Table 9.6.1.1-1: Resource Types**

| **Resource Type** | **Short Description** | **Child Resource Types** | **Parent Resource Types** | **Clause** |
| --- | --- | --- | --- | --- |
| *accessControlPolicy* | Stores a representation of privileges. It is associated with resources that shall be accessible to entities external to the Hosting CSE. It controls "who" is allowed to do "what" and the context in which it can be used for accessing resources | *subscription, transaction* | *AE, AEAnnc, remoteCSE, remoteCSEAnnc, CSEBase, CSEBaseAnnc* | 9.6.2 |
| *AE* | Stores information about the AE. It is created as a result of successful registration of an AE with the Registrar CSE | *subscription, container,* *flexContainer,**group, accessControlPolicy,* *pollingChannel, semanticDescriptor,**timeSeries, transaction, transactionMgmt,**triggerRequest, crossResourceSubscription, backgroundDataTransfer, semanticMashupInstance, locationPolicy, action, multimediaSession, dynamicAuthorizationConsultation, nwMonitoringReq, timeSyncBeacon, primitiveProfile, softwareCampaign* | *CSEBase* | 9.6.5 |
| *container* | Shares data instances among entities. Used as a mediator that buffers data exchanged between AEs and/or CSEs. The exchange of data between AEs (e.g. an AE on a Node in a field domain and the peer-AE on the infrastructure domain) is abstracted from the need to set up direct connections and allows for scenarios where both entities in the exchange do not have the same reachability schedule | *container,* *flexContainer, contentInstance, subscription, latest, oldest，semanticDescriptor, timeSeries, transaction, action* | *AE, AEAnnc, container, containerAnnc, remoteCSE, remoteCSEAnnc,* *CSEBase, CSEBaseAnnc,**flexContainer, flexContainerAnnc* | 9.6.6 |
| *contentInstance* | Represents a data instance in the *<container>* resource | *semanticDescriptor, transaction* | *Container, containerAnnc* | 9.6.7 |
| *flexContainer* | A template which allows to define specialized (customizable) versions of containers with a flexible and lightweight structure  | *container,* *flexContainer, flexContainerInstance, latest, oldest, subscription, semanticDescriptor, timeSeries, transaction, action* | *AE, AEAnnc, container, containerAnnc,* *flexContainer, flexContainerAnnc, remoteCSE, remoteCSEAnnc,* *CSEBase, CSEBaseAnnc,* | 9.6.35 |
| *flexContainerInstance* | Represents a timestamped Data instance in the *<flexContainer>* resource |  *transaction* | *flexContainer* | 9.6.59 |
| *CSEBase* | The structural root for all the resources that are residing on a CSE. Stores information about the CSE itself | *remoteCSE, remoteCSEAnnc, CSEBaseAnnc, node, AE, container, group, accessControlPolicy, subscription, mgmtCmd, locationPolicy, statsConfig, statsCollect, request, delivery,**schedule,**notificationTargetPolicy, semanticRuleRepository,* *flexContainer,**timeSeries, AEContactList, transaction, transactionMgmt, crossResourceSubscription, backgroundDataTransfer, semanticMashupJobProfile, semanticMashupInstance, action, m2mServiceSubscriptionProfile, dynamicAuthorizationConsultation, localMulticastGroup, serviceSubscribedAppRule, authorizationPolicy, authorizationDecision, authorizationInformation, ontologyRepository, nwMonitoringReq, timeSyncBeacon, primitiveProfile, softwareCampaign* | *None specified* | 9.6.3 |
| *delivery* | Forwards requests from CSE to CSE | *subscription, transaction* | *CSEBase* | 9.6.11 |
| *eventConfig* | Defines events that trigger statistics collection | *subscription, transaction* | *statsConfig* | 9.6.24 |
| *execInstance* | Contains all execution instances of the same Management Command | *subscription, transaction* | *mgmtCmd* | 9.6.17 |
| *fanOutPoint (V)* | Virtual resource containing target for group requestIt is used for addressing bulk operations to all the resources that belong to a group | *None specified* | *group* | 9.6.14 |
| *group* | Stores information about resources of the same type that need to be addressed as a Group. Operations addressed to a Group resource shall be executed in a bulk mode for all members belonging to the Group | *fanOutPoint,**subscription,**semanticFanOutPoint, semanticDescriptor, transaction, action* | *AE, AEAnnc, remoteCSE, remoteCSEAnnc, CSEBase, CSEBaseAnnc,* | 9.6.13 |
| *latest (V)* | Virtual resource that points to most recently created *<contentInstance>* , <*timeSeriesInstance*> or <flexContainerInstance> child resource within a *<container>,*  *<timeSeries> or <flexContainer>* resource | *None specified* | *container, timeSeries, or flexContainer* | 9.6.27 |
| *locationPolicy* | Includes information to obtain and manage geographical location. It is only referenced within a container, the *contentInstances* of the container provide location information | *subscription, transaction* | *CSEBase, AE* | 9.6.10 |
| *mgmtCmd* | Management Command resource represents a method to execute management procedures required by existing management protocols | *execInstance,**subscription, transaction* | *CSEBase* | 9.6.16 |
| *mgmtObj* | Management Object resource represents management functions that provides an abstraction to be mapped to external management technology. It represents the node and the software installed in the node (see note) | *subscription, transaction, semanticDescriptor* | *node, mgmtObjAnnc* | 9.6.15Annex D |
| *m2mServiceSubscriptionProfile* | Data pertaining to the M2M Service Subscription | *serviceSubscribedNode,**subscription, transaction, serviceSubscribedUserProfile* | *CSEBase*  | 9.6.19 |
| *node* | Represents specific Node information | *mgmtObj,* *subscription, semanticDescriptor, schedule, transaction, action* | *CSEBase* | 9.6.18 |
| *notificationTargetMgmtPolicyRef* | Represents a list of notification targets and the deletion policy | *subscription, transaction* | *subscription* | 9.6.31 |
| *notificationTargetPolicy* | Represents a notification target deletion policy with pre-defined action and deletion rules | *subscription, policyDeletionRules, transaction* | *CSEBase* | 9.6.32 |
| *notificationTargetSelfReference (V)* | Virtual resource used to remove the Notification Target | *None specified* | *subscription* | 9.6.34 |
| *oldest (V)* | Virtual resource that points to first created *<contentInstance>*, <*timeSeriesInstance*> or *<flexContainerInstance>* child resource within a *<container>, <timeSeries> or <flexContainer>* resource | *None specified* | *container, timeSeries, or flexContainer* | 9.6.28 |
| *pollingChannel* | Represent a channel that can be used for a request-unreachable entity | *pollingChannelURI* | *remoteCSE, AE* | 9.6.21 |
| *pollingChannelURI (V)* | Virtual resource used to perform service layer long polling of a resource Hosting CSE by a request-unreachable entity | *None specified* | *pollingChannel* | 9.6.22 |
| *policyDeletionRules* | Represents a set of rules which is associated with notification target removal policy | *subscription, transaction* | *notificationTargetPolicy* | 9.6.33 |
| *remoteCSE* | Represents a remote CSE for which there has been a registration procedure with the registrar CSE identified by the CSEBase resource | *container, containerAnnc,**contentInstanceAnnc* *flexContainer, flexContainerAnnc,**group, groupAnnc, accessControlPolicy, accessControlPolicyAnnc, subscription, pollingChannel,* *timeSeries,**timeSeriesAnnc, timeSeriesInstanceAnnc,**mgmtObjAnnc,**nodeAnnc,**AEAnnc,**locationPolicyAnnc, transaction, crossResourceSubscription, backgroundDataTransfer, semanticMashupJobProfile, semanticMashupJobProfileAnnc, semanticMashupInstance,**semanticMashupInstanceAnnc,* *action, dynamicAuthorizationConsultation, dynamicAuthorizationConsultationAnnc, nwMonitoringReq, timeSyncBeacon, primitiveProfile, softwareCampaign* | *CSEBase* | 9.6.4 |
| *request* | Expresses/access context of an issued Request | *subscription, transaction* | *CSEBase* | 9.6.12 |
| *schedule* | Contains scheduling information for delivery of messages | *subscription, transaction* | *subscription, CSEBase, node, softwareCampaign* | 9.6.9 |
| *serviceSubscribedNode* | Node information | *subscription, transaction* | *m2mServiceSubscriptionProfile* | 9.6.20 |
| *statsCollect* | Defines triggers for the IN-CSE to collect statistics for applications | *subscription, transaction* | *CSEBase (in IN‑CSE)* | 9.6.25 |
| *statsConfig* | Stores configuration of statistics for applications | *eventConfig,**subscription, transaction* | *CSEBase (in IN‑CSE)* | 9.6.23 |
| *subscription* | Subscription resource represents the subscription information related to a resource. Such a resource shall be a child resource for the subscribed-to resource | *schedule, notificationTargetSelfReference, notificationTargetMgmtPolicyRef, transaction* | *accessControlPolicy, accessControlPolicyAnnc, AE, AEAnnc, container, containerAnnc, CSEBase, delivery, eventConfig, execInstance, group, groupAnnc, locationPolicy, locationPolicyAnnc, mgmtCmd, mgmtObj, mgmtObjAnnc, m2mServiceSubscriptionProfile, node, nodeAnnc, serviceSubscribedNode, remoteCSE, remoteCSEAnnc, CSEBaseAnnc, request, schedule, scheduleAnnc,**semanticDescriptor, semanticDescriptorAnnc, statsCollect, statsConfig,**flexContainer, flexContainerAnnc,**timeSeries, timeSeriesAnnc, nwMonitoringReq, semanticRuleRepository, reasoningRules, reasoningJobInstance, timeSyncBeacon, primitiveProfile,**state,**processManagement* | 9.6.8 |
| *serviceSubscribedAppRule* | Represents a rule that defines allowed App-ID and AE-ID combinations that are acceptable for registering an AE on a Registrar CSE | *subscription, transaction* | *CSEBase* | 9.6.29 |
| *semanticDescriptor* | Stores semantic description pertaining to a resource and potentially sub-resources. | *subscription, transaction* | *AE, container, contentInstance, group, node, flexContainer, timeSeries, mgmtObj* | 9.6.30 |
| *semanticFanOutPoint* | Virtual resource used as target for semantic discovery aimed at a logical graph distributed over multiple *semanticDescriptor* resources, which belong to the corresponding *group* parent resource |  | *group* | 9.6.14a |
| *dynamicAuthorizationConsultation* | Represents consultation information used by a CSE when performing consultation-based dynamic authorization | *subscription, transaction* | *AE, AEAnnc, remoteCSE, remoteCSEAnnc, CSEBase, CSEBaseAnnc,* | 9.6.40 |
| *timeSeries* | Stores and Shares Time Series Data instances among entities. | *timeSeriesInstance, subscription, semanticDescriptor,**latest, oldest, transaction, action* | *AE, AEAnnc, remoteCSE, remoteCSEAnnc, CSEBase, CSEBaseAnnc,**container, containerAnnc, flexContainer, flexContainerAnnc* | 9.6.36 |
| *timeSeriesInstance* | Represents a Time Series Data instance in the *<timeSeries>* resource |  *transaction* | *timeSeries, timeSeriesAnnc* | 9.6.37 |
| *role* | Represents a role that is assigned to an AE or CSE. | *subscription, transaction* | *authorizationInformation* | 9.6.38 |
| *token* | Used for storing a token that is issued to an AE or CSE. | *subscription, transaction* | *authorizationInformation* | 9.6.39 |
| *authorizationDecision* | Represents an access control decision point | *subscription, transaction* | *CSEBase* | 9.6.41 |
| *authorizationPolicy* | Represents an access control policy retrieval point | *subscription, transaction* | *CSEBase* | 9.6.42 |
| *authorizationInformation* | Represents an access control information point | *role**token**subscription, transaction* | *CSEBase* | 9.6.43 |
| *localMulticastGroup* | Stores local multicast group information of member hosting CSE. |  *transaction* | *CSEBase* | 9.6.44 |
| *AEContactList* | Contains information about a CSE that has resources that referencing an AE-ID | *AEContactListPerCSE, subscription, transaction* | *CSEBase* | 9.6.45 |
| *AEContactListPerCSE* | Contains information about a CSE that has resources that referencing an AE resource identifier for tracking purposes | *None specified* | *AEContactList* | 9.6.46 |
| *transactionMgmt* |  | *subscription* | *CSEBase, AE, remoteCSE* | 9.6.47 |
| *transaction* |  | *action* | *All non-virtual resource types with the exception of the following:**request, delivery, pollingChannel, transactionMgmt, transaction, timeSyncBeacon* | 9.6.48 |
| *triggerRequest* | Used by an AE to initiate, replace or recall a device trigger request  | *subscription* | *AE* | 9.6.49 |
| *ontologyRepository* | Represents the collection of the managed ontologies and the semantic validation service | *ontology, semanticValidation, subscription, ontologyMapping, ontologyMappingAlgorithmRepository* | *CSEBase* | 9.6.50 |
| *ontology* | Store the representation of an ontology | *subscription* | *ontologyRepository* | 9.6.51 |
| *semanticValidation* | A virtual resource as the interface to perform semantic validation on the received <semanticDescriptor> resource against the referenced ontology. | *None specified* | *ontologyRepository* | 9.6.52 |
| *ontologyMapping* | Represents a mapping operation between two different ontologies. | *subscription* | *ontologyRepository* | 9.6.70 |
| *ontologyMappingAlgorithmRepository* | Represents the collection of ontology mapping algorithms. | *subscription, ontologyMappingAlogrithm* | *ontologyRepository* | 9.6.71 |
| *ontologyMappingAlogrithm* | Represents an ontology mapping algorithm | *None specified* | *ontologyMappingAlgorithmRepository* | 9.6.72 |
| *semanticMashupJobProfile* | Represents the profile and description of a semantic mashup service | *semanticMashupInstance, semanticDescriptor, subscription* | *CSEBase, remoteCSE* | 9.6.53 |
| *semanitcMashupInstance* | Represents a semantic mashup instance | *semanticMashupResult, semanticDescriptor, mashup, subscription* | *semanticMashupJobProfile, AE, remoteCSE, CSEBase* | 9.6.54 |
| *mashup* | A virtual resource use to trigger the calculation and generation of new mashup result | *Not specified* | *semanticMashupInstance* | 9.6.55 |
| *semanticMashupResult* | Represent semantic mashup results | *semanticDescriptor, subscription* | *semanticMashupInstance* | 9.6.56 |
| *multimediaSession* | Stores a representation of a multimedia session information requested by a registering AE | *subscription*  | *AE* | 9.6.57 |
| *crossResourceSubscription* | represents the cross-resource subscription information related to multiple subscribed-to resources. Such a resource shall include a list of subscribed-to resources as its attribute, or shall be created as a child resource of a <group> resource where member resources shall be the subscribed-to resources.  | *schedule, notificationTargetSelfReference, notificationTargetMgmtPolicyRef, transaction*  | *CSEBase, remoteCSE, AE* | 9.6.58 |
| *backgroundDataTransfer* | Stores information for a background data transfer request | *None specified* | *AE, remoteCSE, CSEBase* | 9.6.60 |
| *action* | Specifies the action(s) that is performed whenever an event is triggered at the <*dependency>* resource | *dependency* | *CSEBase, remoteCSE, node, AE, container, flexContainer, group, timeSeries, transaction, state* | 9.6.61 |
| *dependency* | Specifies the condition(s) of a monitored event which triggers the operation(s) specified by the <*action>* resource | *None specified* | *action* | 9.6.62 |
| *e2eQosSession* | Specifies the end-to-end (E2E) QoS session requirements for the exchange of oneM2M request and response primitives between oneM2M entities. | *subscription*  | *CSEBase, remoteCSE, AE* | 9.6.63 |
| *nwMonitoringReq* | Specifiesed the request that an AE retrieves the Underlying Network information. The resource provides the characteristics of the Underlying Network status in a particular geographic area such as congestion status and number of devices. | *subscription*  | *CSEBase, remoteCSE, AE* | 9.6.64 |
| *semanticRuleRepository* | Specifies one or multiple <reasoningRules> child resources to represent different sets of reasoning rules in the oneM2M system | *reasoningRules, reasoningJobInstance* | *CSEBase* | 9.6.65 |
| *reasoningRules* | Store a set of related reasoning rules | *subscription* | *semanticRuleRepository* | 9.6.66 |
| *reasoningJobInstance* | Represents a specific reasoning job instance for enabling one-time or continuous reasoning operations. | *subscription* | *semanticRuleRepository* | 9.6.67 |
| *serviceSubscribedUserProfile* | Stores user profile information for a given M2M Service User | *subscription, transaction* | *m2mServiceSubscriptionProfile* | 9.6.68 |
| *timeSyncBeacon* | Specifies criteria that a Hosting CSE uses to generate time synchronization beacon notifications  | *subscription,**transaction* | *CSEBase, remoteCSE, AE* | 9.6.69 |
| *primitiveProfile* | Stores message parameters and/or resource attributes that can be applied (i.e. added, removed or modified) to a request or a response primitive.  | *subscription, transaction* | *CSEBase, remoteCSE, AE*  | 9.6.73 |
| *state* | Stores status information and actions for a given state of a process that is managed by a Hosting CSE.  | *subscription, transaction, action* | *processManagement* | 9.6.74 |
| *processManagement* | Specifies a process that a Hosting CSE manages on behalf of an AE that consists of sequence of states defined by <*state*> resources.  | *state, subscription, transaction* | *CSEBase, remoteCSE, AE* | 9.6.75 |
| *softwareCampaign* | Provides policies and rules for the dynamic management of a software by a Hosting CSE.  | *subscription, transaction, schedule* | *CSEBase, remoteCSE, AE* | 9.6.76 |
| NOTE: See clause 9.6.12 for a summary of specializations of *<mgmtObj>.* |

-------------------------------------------------- End of Change 4---------------------------------------