|  |
| --- |
|  |

|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | SDS #39 |
| Source:\* | Convida Wireless |
| Date:\* | 2019-02-19 |
| Contact:\* | Catalina Mladin, Convida, [Mladin.Catalina@convidawireles.com](mailto:Mladin.Catalina@InterDigital.com)  Dale Seed, Convida, [Seed.Dale@convidawireles.com](mailto:Seed.Dale@convidawireles.com) |
| Reason for Change/s:\* | Provides potential solution affecting two Key Issues: KI for service-aware registration and KI for multiple registrations |
| CR against: Release\* | Release 4 |
| CR against: WI\* | Active <WI-0046>  MNT / < Work Item number(optional)>  Is this a companion CR? Yes  No  Companion CR number: (Note to Rapporteur - use latest agreed revision)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\* | TR-0052 V 0.4.0 |
| Clauses/Sub Clauses \* | Clause 9 |
| Type of change: \* | Editorial change  Bug Fix or Correction  Change to existing feature or functionality  New feature or functionality  Only ONE of the above shall be ticked |
| Impacted other TS/TR(s) | <TS/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  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

Several discussions have occurred in oneM2M regarding:

* The need to clarify the concept of “service” in oneM2M, because it is used in requirements and high-level descriptions, but not directly mapped to any architectural entities, resources, etc. This need has been made even more clear by the requirements and features discussed in relationship with the Edge/Fog work, see e.g. ARC-2018-0270-Fog-related\_concepts\_and\_questions. For example, formalizing the concept of “service” is needed in order to be able to define other concepts central to the Edge/Fog topic such as “service orchestration”, “service migration, etc.
* A specific Key Issue on Service Awareness was agreed via ARC-2018-0346R02 which brought together existing requirements, new F/E related usecases and an entire group of existing definitions and concepts which all require clarification and formalization of this concept.
* A TR-0026 Key Issue points to the fact that the requirement for multiple registrations may introduce unnecessary resource allocations at the registrars unless the registration procedure indicates which services are requested or granted, and the specific combinations of services to be provided by each registrar. Support of multiple registrations is required in turn not only by Vehicular Enablement, but also by Edge/Fog related functionality. Furthermore, when requesting service during registration, a registee should also be able to specify the levels of functionality for the services it requires to use.

Note: due to the time elapsed and the extensive changes made for the R02 version of the document, track changes have not been kept, but the all-new text proposal in this version can be easily compared to the previous version, since they target a new document section.

R03 addresses comments received at SDS#39:

* Editor’s notes to clarify that mapping of these concepts to profiles and features is for FFs, in order to be addressed in future contributions
* Added information about <registrationInfo> use

R04 corrects typos

-----------------------Start of change 1-------------------------------------------



# Proposed Solutions

*Editor’s Note: The section provides solutions to the Key Issues identified for employing Edge and Fog technologies*. *in oneM2M*.

## Solution A: Common Service description and service-awareness

*Editor’s Note: Each Solution section references one or more Key Issues that it addresses and provides a brief solution description.*

### Solution Applicability

*Editor’s Note: The Solution Applicability states which Key Issues are addressed by the solution.*

This solution applies to Key Issue X: “Service” awareness and dependencies.

This solution also applies to TR-0026 [i.38] Key Issue 9: Simultaneous registration with multiple CSEs.

### Solution Description

*Editor’s Note: This section provides a concise description of the solution which provides enough detail for further stage 2 development.*

Definitions and concepts

This solution proposes that the concept of M2M Service is refined and clarified in TS-0011 in order to provide a clearer understanding of its relationship with Application Services and Common Services. The changes proposed are also meant to clarify existing relationships (e.g. between M2M Application Service and Application Entity):

The following are existing TS-0011 definitions:

**Application Entity:** represents an instantiation of Application logic for end-to-end M2M solutions

**M2M Application Service:** realized through the service logic of an M2M Application and is operated by the User or an M2M Application Service Provider

**M2M Common Services:** set of oneM2M specified functionalities that are widely applicable to different application domains made available through the set of oneM2M specified interfaces

**Common Services Entity (CSE):** represents an instantiation of a set of Common Service Functions of the M2M environments. Such service functions are exposed to other entities through reference points

**Common Services Function (CSF):** informative architectural construct which conceptually groups together a number of sub‑functions

NOTE: Those sub-functions are implemented as normative resources and procedures. A set of CSFs is contained in the CSE.

**M2M Service:** consists of one or more M2M Application Services and one or more M2M Common Services

The following changes are proposed:

**Application Entity (AE):** represents an instantiation of Application Service logic for end-to-end M2M solutions.

**Application Service (AS):** service logic designed to perform coordinated M2M functions, tasks, or activities for the benefit of the User or an M2M Application Service Provider.

**Common Service (CS):** oneM2M specified functionality that is widely applicable to different application domains, made available through the set of oneM2M specified interfaces.

NOTE: This functionality is implemented as normative resources and procedures. A set of CSs that are functionally related to one another are represented as a CSF.

**Common Services Function (CSF):** represents an instantiation of a set of Common Services that are functionally related to one another.

NOTE: The higher-level functionality of a CSE is described by the set of CSFs instantiated.

**Common Services Entity (CSE):** represents an instantiation of a set of Common Service Functions (CFSs) and their respective Common Services (CSs) in the M2M environment. The CSFs and their respective CSs are exposed to other entities through oneM2M defined reference points.

**M2M Service:** one or more Common Services, Application Services or combination thereof.

This solution also proposes the introduction of type definitions (csfType and csType) and identifiers (CSF-ID and CS-ID) for both Common Services Functions and Common Services. These type definitions and identifiers enable the normative classification, identification and use of CSF and CS instances within the oneM2M system.

Table 9.1.2‑1: Interpretation of csfType

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 100 | Registration (REG) |  |
| 200 | Discovery (DIS) |  |
| 300 | Subscription (SUB) |  |
| 400 | Semantics (SEM) |  |
| 500 | Application and Service Layer Management (ASM) |  |
| 600 | Data Management and Repository (DMR) |  |
| 700 | Device Management (DMG) |  |
| 800 | Group Management (GMG) |  |
| 900 | Transaction Management (TMG) |  |
| 1000 | Communication Management and Delivery Handling (CMDH) |  |
| 1100 | Security (SEC) |  |
| 1200 | Location (LOC) |  |
| 1300 | Network Service Exposure, Service Execution and Triggering (NSSE) |  |
| 1400 | Service Charging and Accounting (SCA) |  |

Table 9.1.2‑2: Interpretation of csType

|  |  |  |
| --- | --- | --- |
| Value | csType | Corresponding csfType |
| 101 | AE Registration | Registration |
| 102 | CSE Registration |  |
|  |  |  |
| 201 | Resource Discovery | Discovery |
| 202 | IPE Discovery |  |
|  |  |  |
| 301 | Resource Subscription | Subscription |
| 302 | Cross Resource Subscription |  |
|  |  |  |
| 401 | Semantic Discovery | Semantics |
| 402 | Semantic Query |  |
| 403 | Semantic Mashup |  |
| 404 | Semantic Reasoning |  |
|  |  |  |
| 501 | Application Management | Application and Service Layer Management |
| 502 | Service Layer Management |  |
|  |  |  |
| 601 | Container | Data Management and Repository |
| 602 | flexContainer |  |
| 603 | Time Series |  |
|  |  |  |
| 701 | Native oneM2M | Device Management |
| 702 | LWM2M Interworking |  |
| 703 | OMA Interworking |  |
| 704 | BBF Interworking |  |
|  |  |  |
| 801 | Unicast Fanout | Group Management |
| 802 | IP Multicast Fanout |  |
| 803 | 3GPP MBMS Fanout |  |
|  |  |  |
| 901 | Creator Controlled | Transaction Management |
| 902 | CSE Controlled |  |
|  |  |  |
| 1001 | Blocking | Communication Management and Delivery Handling |
| 1002 | Non-Blocking Sync |  |
| 1003 | Non-Blocking Async |  |
| 1004 | Flex Blocking |  |
| 1005 | Store-and-Forward |  |
| 1006 | Delivery Aggregation |  |
| 1007 | Polling Channel |  |
|  |  |  |
| 1101 | Enrolment Function (MEF) | Security |
| 1102 | Authentication Function (MAF) |  |
| 1103 | Security Association |  |
| 1104 | Authorization |  |
| 1105 | Dynamic Authorization - Direct |  |
| 1106 | Dynamic Authorization - Indirect |  |
| 1107 | Distributed Authorization |  |
| 1108 | End-to-End Security of Data (ESDATA) |  |
| 1109 | End-to-End Security of Primitives (ESPRIM) |  |
| 1110 | Privacy Policy Management |  |
|  |  |  |
| 1201 | Network Based | Location |
| 1202 | Device Based |  |
| 1203 | Sharing Based |  |
|  |  |  |
| 1301 | Device Triggering | Network Service Exposure, Service Execution and Triggering (NSSE) |
| 1302 | 3GPP SCEF T8 Interworking |  |
|  |  |  |
| 1401 | Information Recording | Service Charging and Accounting |
| 1402 | Offline Charging |  |

Editor’s note: How the CS and CSF concepts, as well as csType and csfType, map to the concepts of product profiles and features is FFS.

Figure 9.1.2-1 shows the logical composition of an example M2M Service composed of several CS and several AS. In this example the relationship between CSs of the same type is shown via the corresponding csfType, with higher granularity than csType, however, either logical construct may be used.



**Figure 9.1.2-1: M2M Service Logical Composition (example)**

The following text reflects the definition of the proposed CSF identifier (CSF-ID):

A CSF is identified by a unique identifier, the CSF-ID, when instantiated within a CSE in the M2M System.

The CSF-ID is unique within the domain of a CSE. It becomes unique within the domain of an M2M Service Provider when prefixed with the CSE-ID. It becomes globally unique when prefixed with the M2M-SP-ID and CSE-ID.

The following text reflects the definition of the proposed CS identifier (CS-ID):

A CS is identified by a unique identifier, the CS-ID, when instantiated within a CSF in the M2M System.

The CS-ID is unique within the domain of a CSF. It becomes unique within the domain of a CSE when prefixed with the CSF-ID. It becomes unique within the domain of an M2M Service Provider when prefixed with the CSE-ID and CSF-ID. It becomes globally unique when prefixed with the M2M-SP-ID, CSE-ID and CSF-ID.

Figure 9.1.2-2 shows an example instantiation of a M2M Service composed of several CS and several AS. The figure provides an example implementation using several oneM2M entities (CSEs and AEs) with SP-relative IDs. Note that the one-to-one mapping between an AS and its instantiation as an AE allows AS identification to be provided by the corresponding AE-ID and be uniquely identified within the system. Note that the proposed CS-ID and CSF-ID schemes also allow for unique identification, as well as mapping to the corresponding M2M entities instantiating these services.



**Figure 9.1.2-2: IDs for an example M2M Service Instantiation with multiple component Common and Application Services**

#### Impacted Resources and Attributes

##### Overview

To implement this solution, several resources need to be modified, and some new resource types need to be defined These are further detailed in the following clauses.

##### Modified <*CSEBase*> resource

New attributes for the <CSEBase> resource are proposed as shown in the table below.

**Table 9.1.2.2.2-1: New attributes of *<CSEBase>* resource**

| **Attributes of  *<remoteCSE>*** | **Multiplicity** | **RW/**  **RO/**  **WO** | **Description** |
| --- | --- | --- | --- |
| *supportedCommonServices* | 0..1(L) | RW | This attribute contains a list of the types of Common Services Functions (CSFs) and their respective types of Common Services (CS) supported by the CSE. |
| *CommonServiceLevelsSupported* | 0..1(L) | RW | This attribute contains a list of service level requirements organized based on each individual service supported by the CSE. For example, Service Request Rate Requirement, Service Data Storage Requirement, Service Response Delay Requirement and Service Reliability Requirement. |

##### Modified <*remoteCSE*> resource

New attributes for the <remoteCSE> resource are proposed as shown in the table below.

**Table 9.7.2.2.3-1: New attributes of *<remoteCSE>* resource**

| **Attributes of  *<remoteCSE>*** | **Multiplicity** | **RW/**  **RO/**  **WO** | **Description** | *<remoteCSEAnnc> Attributes* |
| --- | --- | --- | --- | --- |
| *requestedCommonServices* | 0..1(L) | RW | This attribute contains a list of the types of Common Services Functions (CSFs) and their respective types of Common Services (CS) requested by the Registree CSE. | OA |
| *grantedCommonServices* | 0..1(L) | RW | This attribute contains a list of the CSF-IDs and CS-IDs that the Registree CSE is granted access to use. | OA |
| *CommonServiceLevelRequirements* | 0..1(L) | RW | This attribute contains a list of service level requirements organized based on each individual service required by the Registree. For example, Service Request Rate Requirement, Service Data Storage Requirement, Service Response Delay Requirement and Service Reliability Requirement. | OA |

##### Modified <*AE*> resource

New attributes for the <AE> resource are proposed as shown in the table below.

**Table 9.1.2.2.4-1: New attributes of *<AE>* resource**

| **Attributes of  *<AE>*** | **Multiplicity** | **RW/**  **RO/**  **WO** | **Description** | *<AEAnnc> Attributes* |
| --- | --- | --- | --- | --- |
| *requestedCommonServices* | 0..1(L) | RW | This attribute contains a list of the types of Common Services Functions (CSFs) and their respective types of Common Services (CS) requested by the Registree AE. | OA |
| *grantedCommonServices* | 0..1(L) | RW | This attribute contains a list of the CSF-IDs and CS-IDs that the Registree AE is granted access to use. | OA |
| *CommonServiceLevelRequirements* | 0..1(L) | RW | This attribute contains a list of service level requirements organized based on each individual service required by the Registree. For example: Service Request Rate Requirement, Service Data Storage Requirement, Service Response Delay Requirement and Service Reliability Requirement. | OA |

##### Modified <*node*> resource

New child resources as shown in the table below.

**Table 9.1.2.2.5-1 New Child Resources of <*node*>**

| **New Child Resources of *<CSEBase>*** | **Child Resource Type** | **Multiplicity** | **Description** |
| --- | --- | --- | --- |
| *[variable]* | <registrationInfo> | 0.. n | This child resource stores the registration information associated with the hosted CSE or AEs. |

##### New Resource Type: <*registrationInfo>*

The <*registrationInfo*> resource is proposed as a child of the <node> resource to store registration information of the CSEs and AEs hosted on the node. One <*registrationInfo*> resource is used for keeping a record of the set of common services granted to a Registree from each of its Registrar CSEs and is available to the entities hosted in the node or to be discovered by other entities.

For example, the <*node*> resource of an ASN with multiple AEs will contain <*registrationInfo*> with entries for the hosting CSE as well as each of the AEs hosted on that node. In addition, the <*registrationInfo*> resource in this case will include information about the registrations of registree ADNs. This allows information knowledge of the various registrations (e.g. of the ASN-CSE) to be available locally, e.g. to determine which registrar to use for what functionality.

**Table 9.1.2.2.6-1:**  **Attributes of *<registrationInfo>* resource**

| **Attributes of  *< registrationInfo*>** | **Multiplicity** | **RW/**  **RO/**  **WO** | **Description** |
| --- | --- | --- | --- |
| *registree* | 1 | RW | This attribute contains the identifier of the Registree, i.e. either CSE-ID or AE-ID |
| *registrationList* | 1(L) | RW | A list providing information for each registration. Each member of the list comprises of tuples as described in table 9.1.2.2.6-2, . |

The information represented in the *registrationList* comprises of tuples (*registrar*, *grantedCommonServices* ) with parameters shown in table 9.1.2.2.6-2

Table 9.1.2.2.6-2: Parameters in registration-tuples

| Name | Description |
| --- | --- |
| *registrar* | CSE-ID of the Registrar CSE |
| *grantedCommonServices* | A list of the CSF-IDs and CS-IDs that the Registree AE is granted access to use |

#### Impacted Information Flows

##### Registration

To implement this solution, the AE Registration will be affected as follows:

* AE registration procedure (oneM2M TS-0001 [i.3] clause 10.2.2.2):
  + At AE registration, the AE provides the <AE> resource attributes, including *requestedCommonServices* and *commonServiceLevelRequirements*.. Based on these attributes, the Registrar CSE determines if the registration is accepted by checking if the requested services are supported and also if the node that the Registrar is hosted on has available system level resources to meet the service requirements requested by the Registree AE. If the registration request is accepted, the Registrar CSE creates the <AE> resource with the given attributes and in addition it populates the *grantedCommonServices* attribute with the CSF-IDs and CS-IDs of the granted services.
  + Upon receiving a successful registration response, if the registration is a first successful registration of the AE, the initiating AE creates a corresponding <*registrationInfo>* resource as child of the <*node*> resource, capturing its AE-ID in the *registree* attribute. The initiating AE also updates the *registrationList* with the CSE-ID of the Registrar and *grantedCommonServices* from the newly created *<AE>* resource. If the initiating AE already has existing registrations, it adds a new registration-tuple to the *registrationList* attribute of the existing <*registrationInfo>* resource and configures the registration-tuple similarly to the new registration case.
* At CSE registration procedure (oneM2M TS-0001 [i.3], clause 10.2.2.7):
  + At CSE registration, the CSE provides the <remoteCSE> resource attributes, including *requestedCommonServices* and *commonServiceLevelRequirements*. Based on these attributes, the Registrar CSE determines if the registration is accepted by checking if the requested services are supported and also if the node that the Registrar is hosted on has available system level resources to meet the service requirements requested by the Registree CSE. If the registration request is accepted, the Registrar CSE creates the <remoteCSE> resource with the given attributes and in addition it populates the *grantedCommonServices* attribute with the CSF-IDs and CS-IDs of the granted services.
  + Upon receiving a successful registration response, if the registration is a first successful registration of the CSE, the initiating CSE creates a corresponding <*registrationInfo>* resource as child of the <*node*> resource, capturing its CSE-ID in the *registree* attribute. The initiating CSE also updates the *registrationList* with the CSE-ID of the Registrar and *grantedCommonServices* from thenewly created *<remoteCSE>* resource. If the initiating CSE has existing registrations, it adds a new registration-tuple to the *registrationList* attribute of the existing <*registrationInfo>* resource and configures the registration-tuple similarly to the new registration case.

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

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.?