|  |  |
| --- | --- |
| Input Contribution | |
| Meeting ID\* | SDS#40 |
| Title:\* | TR-0059-platform\_discovery\_key\_issue |
| Source:\* | JaeSeung Song, KETI, [jssong@sejong.ac.kr](mailto:jssong@sejong.ac.kr)  Youngjin Na, Hyundai Motors, [yjra@hyundai.com](mailto:yjra@hyundai.com)  Minbyeong Lee, Hyundai Motors, [minbyeong.lee@hyundai.com](mailto:minbyeong.lee@hyundai.com) |
| Date:\* | 2019-05-10 |
| Input related to\* | TR-0059 Services and Platform Discovery |
| Intended purpose of  document:\* | Decision  Discussion  Information  Other <specify> |
| Impacted other TS/TR(s) | N/A |
| Decision requested or recommendation:\* | Introducing Key Issue regarding oneM2M platform discvoery |
| 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

**1. Motivation:**

Now a days oneM2M systems are used in many smart city projects and trials. Let us consider a smart city having a large number of oneM2M platforms (i.e., IN-CSEs from different vendors and service providers) providing various IoT services via various IoT service providers. Then we can easily come up with the following platform discovery issues:

* How to find IN-CSEs that an IoT application wants to use among a large amount of IoT service platforms in a city?
* In some cases, IoT applications do not necessarily need to know which provider provides which service. There are plenty of public IoT services from different IoT platforms (e.g., smart parking platforms supported by a local government)
* What if Alice (from SanDiego) visits Seoul. Could Alice discovers available IN-CSEs supporting smart parking with her oneM2M smartphone application? (platforms and services should be discoverable)
  + Alice may want to discovery platforms then uses available services from the selected oneM2M platform.
  + Alice may want to discovery a specific IoT service (e.g., smart parking) from where she is located in.
* Could IoT applications know which IN-CSEs provide service for V2X or Smart Parking?
* Do we know how many oneM2M IoT service platforms are running over the world?

**2. Proposed Solutions:**

This contribution includes a proposed solution for oneM2M platform discovery and service provisioning via visited oneM2M CSE. The solution is based on a registry entity managing available oneM2M CSEs. The solution involves a oneM2M CSEs performing registration to a oneM2M DNS-like registry server. By performing these procedures a oneM2M entity can discover available oneM2M platforms whereever it is located in.

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

### 6.2.x Key Issues on oneM2M platforms discovery and local service provisioning

Discovery of oneM2M Platforms is the capability for oneM2M applications to discovery available oneM2M IoT service platforms. Similar to other platforms discovery, in order to support oneM2M platform discovery, a registry entity can be introduced to manage oneM2M platforms. An oneM2M service platform can be published to the registry with its description such as point of access, deployed location, supporting IoT services, supporting oneM2M features. oneM2M platform discovery can be based on such description (e.g., discover oneM2M platforms at Seoul, discover oneM2M platforms supporting smart parking services, discovery oneM2M platforms supporting MQTT). Once a oneM2M platform is discovered, a oneM2M application can register itself to the discovered oneM2M platform to receive required oneM2M services. In addition, oneM2M services offered by the Home Registrar CSE can be supported via the newly registered Visited Registrar CSE. The registry entity can check the liveness of registered oneM2M platforms to provide properly working oneM2M platforms.

The following key issues may be summarized:

1. The oneM2M System currently does not support methods to allow oneM2M entities to discovery available oneM2M platforms
2. The oneM2M System should be able to check the liveness of other oneM2M platforms so that reliability of oneM2M services and platforms discovery can be increased.

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

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

## 8.X Solution: oneM2M Platform Discovery

### 8.X.1 Overview

To enable a oneM2M entity to discover available oneM2M platforms based on certain criteria (such as location, supporting services), this solution proposes to introduce a Registry server managing oneM2M platforms. This solution is ideal for the case where multiple oneM2M platforms are available in a smart city or state or a coutry, so that a oneM2M entity can select a required oneM2M platform even it does not have any prior information about available oneM2M platforms. The Registry and registered oneM2M platforms guarantee that only properly working oneM2M platforms are registered to the registry via liveness checking.

### 8.X.2 Solution Description using Registry

### This clause describes the procedure for a central registry based oneM2M platform discovery. In order to discover oneM2M platforms, a service registry managing available running oneM2M platforms is required. A proper description of oneM2M service platforms and registration procedures are needed to be defined.



**Figure 8.X.2-1: High-level procedure for a Registry based oneM2M platform discovery**

Service platform discovery is the process of locating IoT service platforms and retrieving IoT service platform descriptions that have been previously registered. Interrogating platforms involve querying the service registry for IoT service platforms matching the needs of a service platform requestor.The description of oneM2M platforms can include information as follows:

* Contact of Address (IP Address)
* Port number
* Name of IN-CSE
* Status
* Location
* Profile of IN-CSE
* Type of IN-CSE
* Supporting public services
* Maintenance information (for example, from 01:00 ~ 02:00)
* Access information (or credencial)

After the discovery process is complete, the IoT application can know the exact location of a needed oneM2M platform via CoA, its capabilities, and how to communicate with it.

The registry for IoT service platforms is providing a smilar service to App-ID registry. The CoA of the central Registry can be pre-provisioned.

As there exist oneM2M platforms which are not available because of various reasons such as maintenance, out of order and temporary disorder, the registry has to check the liveness of the registered oneM2M platforms periodically.

### 8.X.3 Solution Description using <*platformRegistry*> resource

### This clause describes the procedure for oneM2M platform discovery based on introducing <platformRegistry> resource. In order to discover oneM2M platforms, the <platformRegistry> resource can be used to keep all the available oneM2M platforms either open to public or have business relationship. For example, if a citizen from smart city A visits smart city B to buy a new temperature sensor, the citizen may want to find out available parking lots using oneM2M smart parking application. In this case, the oneM2M smart parking application can discover available oneM2M platforms supporting smart parking service in smart city B via looking into the <platformRegistry> resource.

In this case, synchronization between the attributes of remote CSEs in the <platformRegistry> resource and the original resource (i.e., available oneM2M platforms) shall be the responsibility of the <platformRegistry> resource hosting CSE. In addition, the hosting CSE performs the liveness check for the all the remoteCSEs managed in the <platformRegistry> resource.

An AE or other CSE can request the source CSE for adding its information to the <platformRegistry> resource of the target Hosing CSE.



Figure 8.x.3-1: Procedure for creating a new platform description record to the <platformRegistry> resource

Step 001: The Originator of a Request initiating the publication of the description of a platform to a target CSE. The request should include the target CSE address and the indication of publishing platform description. The target CSE can also be the Originator of a Request.

Step 002: The Hosting CSE then prepares a CREATE request message of itself to the given target CSE’s <platformRegistry> resource. The message is composed of the address of itself, access tocken to be used for a basic authentication, supporting services and features.

Step 003: The Hosting CSE send the CREATE request message to the target CSE

Step 004: The Target CSE adds a new platform desceiption to the <platformRegistry> resource

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