|  |
| --- |
| CHANGE REQUEST |
| Meeting ID:\* | TDE 54.1 |
| Source:\* | Bob Flynn (Exacta GSS); bob.flynn@exactagss.com |
| Date:\* | 30 May 2022 |
| Reason for Change/s:\* | Add material for time series capabilities  |
| CR against: Release\* | Rel-5 |
| CR against: WI\* | [x]  Active <WI-0107> [ ]  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\* | TR-0057v0.6.0 |
| Clauses \* | 7.4 |
| 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.

# Introduction

While working on “Developer Guide: Semantics” it became clear that some updates to this TR are justified.

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

### 7.4 Semantics

The assumption of many existing oneM2M applications is that they interact with other oneM2M applications through known resource structures. They either create the resources themselves or are configured to use specific resources. Information is typically stored in containers, sometimes as base64-encoded content instances, with the implicit assumption that applications have a-priori knowledge of the syntax and semantics of this information.

Depending on a-priori knowledge of the structures and data works well for small-scale vertical deployments of IoT devices. When the deployment evolves to include new devices, the existing applications change to reflect the new additions. However, in larger systems of IoT devices where the IoT devices may be a part of a legacy deployment or more than a single vertical solution, changes to all existing applications may become impractical.

Using semantic annotations in oneM2M aspects of IoT devices can be described using RDF triples, which is a standard semantic format. The vocabulary used for a semantic description can be defined according to an ontology such as SAREF. Using semantic discovery, applications can describe precisely what information they need or can deal with. This is powered by specifying a semantic filter using the SPARQL query language. The SPARQL filter is matched against the respective semantic annotations of each resource within the discovery scope. This feature in oneM2M helps applications to properly handle the data from the IoT devices.

The oneM2M Base Ontology can be used to describe the resource structure used to model devices and to make them interoperable if the appropriate semantic annotations are made and semantic filtering is used for discovery. This example scenario describes a clothes washing machine and an application to monitor and control the IoT enabled product.



Figure 7.4‑1: Functional Architecture for Smart Clothes Washing Machine

We can use the SAREF ontology to describe the services of the washing machine and the oneM2M Base Ontology to describe the oneM2M interface for the services.

The services of any clothes washing machine are fundamentally the same regardless of how it is odelled in the oneM2M resource tree. The following RDF triples describe the services and functions of our clothes washing machine.

Table 7.4‑1: semantic annotation of a washing machine in oneM2M using SAREF ontology

|  |
| --- |
| sn:WASH\_XYZ a <http://www.XYZ.com/WashingMachines#XYZ\_Cool> ; rdfs:comment “Very cool Washing Machine” ; saref:hasFunction sn:WASH\_XYZ-MonitoringFunction , sn:WASH\_XYZ-StartStopFunction ; saref:hasManufacturer “XYZ” ; saref:hasService sn:WASH\_XYZ-MonitorService , sn:WASH\_XYZ-SwitchOnService ; saref:hasState sn:WASH\_XYZ-WashingMachineStatus ; s4bldg:isContainedIn sn:My\_Bathroom .sn:WASH\_XYZ-StartStopFunction-OFF\_Command a saref:OffCommand .sn:WASH\_XYZ-StartStopFunction-Toggle\_Command a saref:ToggleCommand .sn:WASH\_XYZ-StartStopFunction-ON\_Command a saref:OnCommand .sn:WASH\_XYZ-MonitoringFunction a saref:SensingFunction ; saref:hasCommand sn:WASH\_XYZ-MonitoringFunction-WashingMachineStatus .sn:WASH\_XYZ-StartStopFunction a saref:ActuatingFunction ; saref:hasCommand sn:WASH\_XYZ-StartStopFunction-Toggle\_Command , sn:WASH\_XYZ-StartStopFunction-OFF\_Command ,  sn:WASH\_XYZ-StartStopFunction-ON\_Command . |

With the previous semantic descriptors in place, the following query can be used to find all services offered by washing machines of manufacturer XYZ.

SELECT distinct ?operation ?wm where {

 ?wm a sn:WASH\_XYZ ;

 ?wm oneM2M:hasService ?service ;

 ?service oneM2M:hasOperation ?operation .

}

Once the desired devices are located additional queries can be made to identify exactly how to use the discovered services.

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

### -----------------------Start of change n---------------------------------------------

### -----------------------End of change n---------------------------------------------