|  |
| --- |
|  |

|  |
| --- |
| Input Contribution |
| Meeting ID\* | RDM#56 |
| Title:\* | Key issue on sensor calibration |
| Source:\* | JaeSeung Song, Sejong University, jssong@sejong.ac.kr  |
| Date:\* | 2022-09-28 |
| Input related to\* | TR-0068 (Rel-5) |
| Intended purpose ofdocument:\* | [x]  Decision[ ]  Discussion[ ]  Information[ ]  Other <specify> |
| Impacted other TS/TR(s) |  |
| Decision requested or recommendation:\* | Agree for inclusion in TR-0068 |
| 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 introduces a key issue on sensor calibration.

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

## 8.x Key Issue & Possible Concept x – Sensor calibration

### 8.x.1 Key Issue

The basic concept of the use case introduced in Clause 7.7 is to use Machine Learning in a situation where continuous IoT device calibration is required. In order to support this use case, the IoT platform is required to perform machine learning to generate a calibration value for an IoT device using data collected for a certain period from reference devices. The IoT platform then uses the output from Machine Learning to calibrate the target IoT device. (Optionally, the target device can download the output calibration value into its local memory and do calibration in the device) As the IoT device requires calibration regularly or when its measurement deviates from the standard value, the IoT platform can continuously perform Machine Learning for calibration.

Currently, the oneM2M platform does not provide any features supporting AI/ML and calibrating deviated measurement data. If the oneM2M platform support such sensor calibration features, various IoT service players which require accurate measurement can easily provide high performance and quality service.

### 8.x.2 Possible Solution

Possible solution should support the following requirements (see Clause 7.7):

1. The oneM2M System shall be able to manage calibration information and training datasets for ML to eliminate or minimize measurement errors from IoT sensors.
2. The oneM2M System shall be able to perform ML using training datasets from reference IoT devices and notify calibration results to a target sensor that requires calibration.

This can be done by introducing a set of attributes to a new resource (for example, <*devCalibration*>) or the <*flexContainer*> resource to hold the information required to perform continuous sensor calibration using AI/ML. The required attributes are as follows:

* *calibrationInterval*: defines intervals to perform machine learning for calibration
* *refCalDevices*: list of reference devices to perform ML for calibration. IoT platform uses the value of these referenced devices as training data for calibration
* *mlModel*: defines machine learning model for calibration. IoT platform aware which model to use to perform machine learning for calibration.
* *calibrationLogs*: provides information when performed calibration previously.
* *calibrationValue*: stores the result of machine learning for calibration. IoT device uses this value for calibration until next calibration
* *standardValue*: defines acceptable values for the measurements

Wich such attributes, the oneM2M IoT platform can support calibrating data measurements that drift over time.



Figure 8.x-1: An example structure of [*devCalibration*] resource

In this case, an IoT platform can be considered to have devices that require data calibration and send their measurement to the IoT Platform continuously. When the calibration time interval reaches based on the *calibrationInterval* attribute, or the measurement deviates from the standard range stored in the *standardValues* attribute, the IoT platform performs AI/ML based on the model stored in the *mlModel* attribute. Then the IoT platform starts AI/ML using the collected training data from the referring high-accuracy temperature sensors based on the *refCalDevices* attribute. The IoT platform generates an updated calibration value and stores it in the *calibrationValue* attribute. IoT devices that require this updated calibration value get notifications about it and use it to generate calibrated measurements.

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