oneM2M
Technical Specification

| | |
|---|---|
| Document Number | TS-0041-V-5.4.0 |
| Document Name: | oneM2M-SensorThings Interworking |
| Date: | 2025-08-04 |
| Abstract: | oneM2M TS-0041 defines the interworking specification between the oneM2M service layer and the OGC SensorThings API to enable seamless integration of IoT data and services, particularly in smart city environments. |

Template Version: January 2020 (Do not modify)



About oneM2M

The purpose and goal of oneM2M is to develop technical specifications which address the need for a common M2M Service Layer that can be readily embedded within various hardware and software, and relied upon to connect the myriad of devices in the field with M2M application servers worldwide.

More information about oneM2M may be found at: http//www.oneM2M.org



Notice of Disclaimer & Limitation of Liability

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. NO oneM2M PARTNER TYPE 1 SHALL BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY THAT PARTNER FOR THIS DOCUMENT, WITH RESPECT TO ANY CLAIM, AND IN NO EVENT SHALL oneM2M BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. oneM2M EXPRESSLY ADVISES ANY AND ALL USE OF OR RELIANCE UPON THIS INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

# Contents

# 1 Scope

The present document provides the interworking specification between the oneM2M service layer and the OGC SensorThings API to enable seamless integration of IoT data and services, particularly in smart city environments.

# 2 References

References are either specific (identified by date of publication and/or edition number or version number) or nonspecific. For specific references,only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

## 2.1 Normative references

- [1] OGC SensorThings API "Part 1: Sensing Version 1.1" (http://www.opengis.net/doc/is/sensorthings/1.1
- [2] oneM2M TS-0033 (V3.0.0): "Interworking Framework"
- [3] oneM2M TS-0001 (V4.23.0): "Functional Architecture"

## 2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

- [i.1] oneM2M Drafting Rules (http://www.onem2m.org/images/files/oneM2M-Drafting-Rules.pdf)

# 3 Definition of terms, symbols and abbreviations

## 3.1 Terms

## 3.2 Symbols

## 3.3 Abbreviations

# 4 Conventions

The key words “Shall”, “Shall not”, “May”, “Need not”, “Should”, “Should not” in this document are to be interpreted as described in the oneM2M Drafting Rules [i.1]

# 5 Introduction to OGC SensorThings API

The SensorThings API (STA) is a standard of the Open Geospatial Consortium (OGC). It provides a framework for communication and exchanging data between sensors and applications. The standard is devided in two parts. SensorThings API Part 1 is dedicated to sensing and was published in 2016 and updated in 2021.
A STA-based architecture works in client/server mode. A sensor device pushes data to the SensorThings Server via HTTP. A SensorThings Server may also support MQTT protocol to support publish and subscribe capabilities. An interested application can subscribe to the MQTT-Broker, in order to get notified about new sensor events.

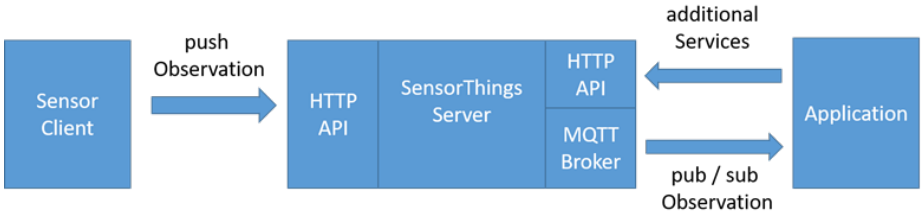


Figure 1: Figure 5-1 STA message flow

The data in the SensorThings server are organized as according to **Sensing Entities** (see Figure 5-2: Sensing Entities data model.

In the Sensing Entities Data Model events or sensor data are called “observations”. Before a sensor is able to push an observation to the server it needs at least a ‘Thing’ and a ‘Datastream’ entity. This has to be created beforehand. One ‘Thing’ might have different ‘Sensors’, one ‘Location’ or many ‘HistoricalLocations’.

The Sensing Entities data model and the purpose of data within the data model discloses mainly two data characteristics, associated with a ‘thing’: - Data

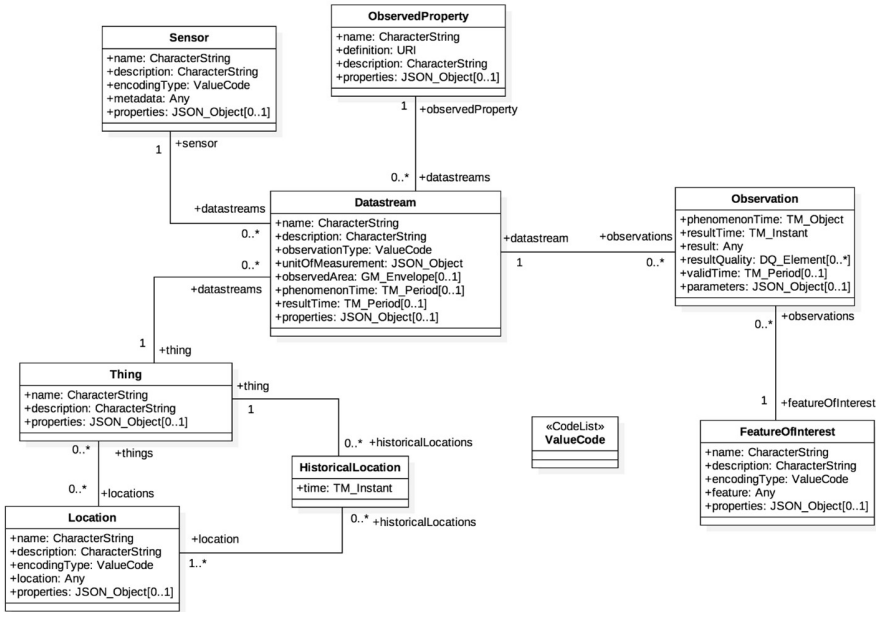


Figure 2: Figure 5-2 STA Sensing Entities Data Model

observations originated by sensors or commands sent to interact with actuators may be seen as IoT data from oneM2M point of view. While: - Data embedded in the Sensing Entities Data Model, like “historic locations” should be seen as data for documentation purposes.

# 6 Architecture Model of OGC/STA to oneM2M interworking

## 6.0 Overview

Figure 6.0-1 shows an architecture approach for an Interworking Proxy Entity (IPE) between oneM2M and the OGC SensorThings API. The IPE is located between a oneM2M CSE and an OGC/SensorThings API (STA)-Server.

The basic interworking enables applications that are connected to an oneM2M-based system to get data from sensors that are connected to an OGC/STA server. Furthermore, an application that is connected to an OGC/STA server will be able to get data from sensors that are connected to an oneM2M-based system. The communication flow of the IPE shall rely on HTTP and MQTT. The MQTT protocol enables publish-subscribe functionality for the OGC side, as specified in the MQTT extension of the SensorThings API [i.1].

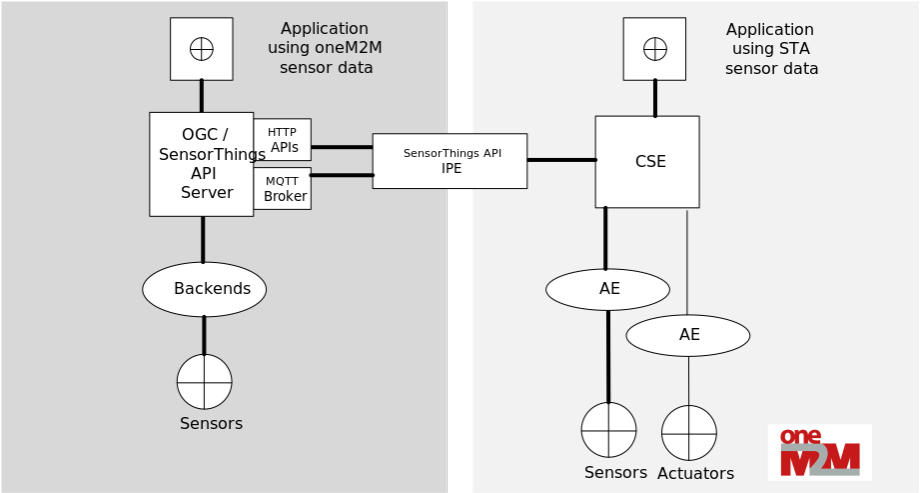


Figure 3: Figure 6.0-1: IPE architecture overview

## 6.1 OGC/STA-to-oneM2M Data Model Mapping

According to oneM2M TS-0033 [2] a representation of a non-oneM2M Proximal IoT function/device in a oneM2M-specified resource instance is to be synchronized with the entity that it represents. Thus the OGC/STA data model has to be represented in the hosting CSE. The SensorThings data model is comprehensive and should be regarded as a n:m relational database structure, holding both: - sensor (IoT-data); and - administrative data (like historic locations or historic products IDs).

The IPE shall map the 'result' attribute of an OGC/STA 'Observation' to the 'content' attribute of a oneM2M *<contentInstance>* resource, and vice versa as shown in Figure 6.1-1. The data type of the 'result' field of an "Observation" is according to SensorThings API [i.1] 'any' and depends on the 'observationType' defined in the associated "Datastream". The 'content' attribute of an oneM2M instance may be stringified data [3] understandable with the help of the 'contentInfo' attribute. The 'contentInfo' attribute on the oneM2M side may be added by the IPE. The original timestamps, present in the "Observation" as 'phenomenonTime' and in the *<contentInstance>* resource as "creationTime," shall be discarded. These timestamps are to be reset by the OGC /STA server and the CSE. They may be transmitted for informational purposes as part of the 'result' or the 'content' fields.

## 6.2 Communication Flow

Figure 6.2-1 shows the oneM2M-to-OGC/STA direction. In order to transfer data from a oneM2M sensor to the OGC/STA server the IPE creates a *<subscription>* resource to the *<container>* resource in the CSE contain-

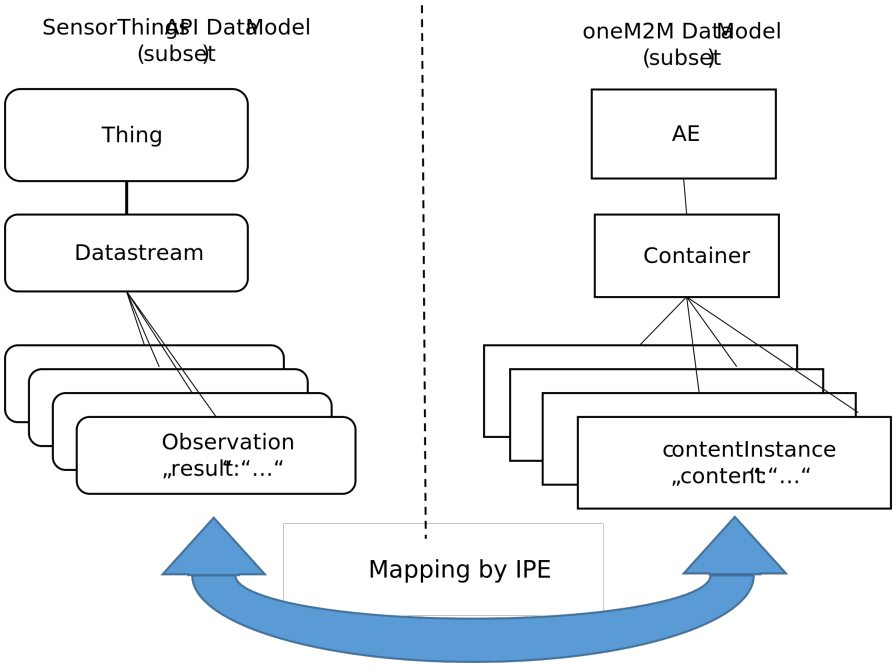


Figure 4: Figure 6.1-1: OGC / STA-to-oneM2M data model mapping

ing the desired data. Triggered by a sensor event a new *<contentInstance>* resource is added to the *<container>* resource by the *<AE>* resource. The IPE gets a notification containing the *<contentInstance>* resource. The IPE constructs an “Observation” creation request and copies the ‘content’ attribute of the *<contentInstance>* resource to the ‘result’ attribute of the “Observation” and sends it to a “Datastream” to be created as detailed in Section 6.3.1 at the OGC/STA server. The OGC/STA applcation gets the sensor data either by polling the OGC/STA server or subscribing to the corresponding “Datastream” at the MQTT broker of the OGC/STA server.

Figure 6.2-2 shows the OGC/STA-to-oneM2M direction. The IPE subscribes to the desired “Datastream” of the MQTT-Broker at the OGC/STA server. The OGC/STA server publishes a new “Observation” via the MQTT broker triggered by a OGC/STA sensor. The IPE creates a *<contentInstance>* resource in a container, to be created as detailed in Section 6.3.2 in the CSE and copies the ‘result’ attribute of the “Observation” to the ‘content’ attribute of the *<contentInstance>* resource. The oneM2M applcation gets the sensor data either by polling the CSE or subscribing to the desired *<container>* resource at the CSE.

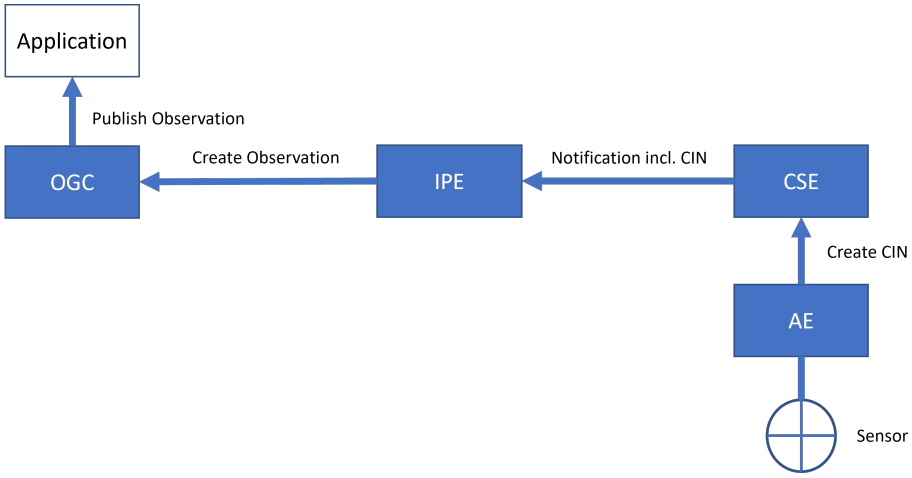


Figure 5: Figure 6.2-1: Communication oneM2M-to-OGC/STA direction

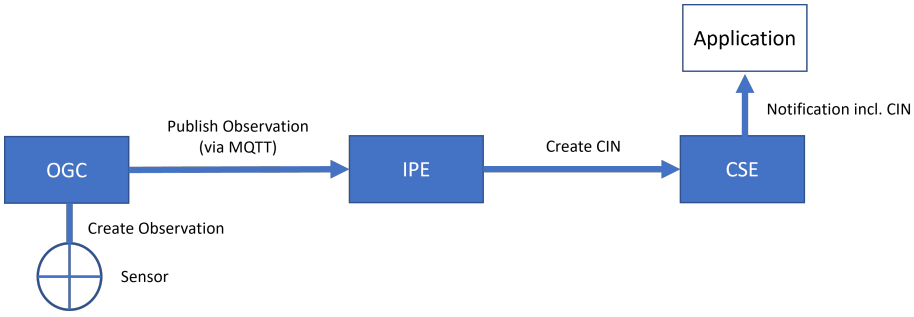


Figure 6: Figure 6.2-2: Communication OGC/STA-to-oneM2M direction

## 6.3 Configuration Aspects

### 6.3.0 Introduction

To enable interworking, preparation is required for both the oneM2M-CSE and the OGC/STA server (see Figure 6.3.0-1). As described in Section 6.0, the IPE maps data from an OGC/STA "Observation" to a oneM2M *<contentInstance>* resource and vice versa. This specification defines a 1-to-1 relationship in each direction between the "Datastream" associated with the "Observation" and the *<container>* resource associated with the *<contentInstance>* resource. An IPE may implement multiple 1-to-1 relationships.


Figure 7: Figure 6.3.0-1: Both sides of the IPE configuration

### 6.3.1 Configuration of OGC/STA Server Side

**6.3.1.0 Overview** Both directions of the data flow between the OGC/STA server and the IPE require their own configuration steps.

**6.3.1.1 Communication direction OGC/STA Server towards IPE** In Figure 6.3.1.1-1, an OGC/STA client is connected to an OGC/STA server, and its data is forwarded to the IPE. The OGC/STA client publishes data to the OGC/STA server via an HTTP-POST message.

An "Observation" according to STA Sensing Entities Data Model [i.1] belongs to a "Datastream" (see Figure 5-2). The IPE shall subscribe to the "Datastream" containing the observations to be forwarded to the oneM2M side at the MQTT broker of the OGC/STA server using its specific URL or topic, e.g., {sta-example-server-address.com/v1.0/Datastreams(8715)}. Upon successful subscription, the IPE will receive every "Observation" pushed to that "Datastream".

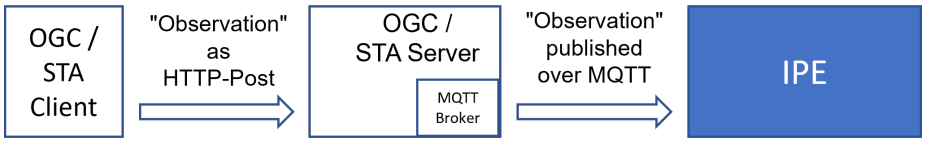


Figure 8: Figure 6.3.1.1-1: Message flow from OGC/STA Client to OGC/STA Server to IPE

**6.3.1.2 Communication direction IPE towards OGC/STA Server** The IPE requires a destination-"Datastream" to send an "Observation" containing data from the oneM2M side. If no associated "Datastream" exists on the OGC/STA server, it shall be created. This can be done beforehand or at the IPE's start-up, depending on the implementation. When a "Datastream" is created on the OGC/STA server, a reference ID (e.g. {"@iot.id:3635353"}) is returned. This reference is required by the IPE to associate an "Observation" with a "Datastream" and shall be available during IPE operation. In addition to the "Datastream" other entities of the STA Sensing Entities Data Model [i.1], such as "Location" or "Sensor," may be created.

The creation of entities like "Datastream" and "Thing" requires several mandatory properties that shall be known at configuration time (e.g., 'name' and 'description'). These property fields may be automatically derived, for example, from the "Label" or "ResourceName" attributes of the corresponding oneM2M *<container>* resource or if existing, from the corresponding *<AE>* resource during IPE configuration. The OGC/STA procedures for creating OGC entities are described in SensorThing API documentation [i.1].

Once the destination-"Datastream" is created, the IPE can send an "Observation" to the OGC/STA server as HTTP POST message. An interested OGC/STA client can subscribe to the destination-"Datastream" at the MQTT Broker of the OGC/STA server to receive each "Observation" forwarded by the IPE (see Figure 6.3.1.2-1). Alternatively, the OGC/STA client may use an HTTP-GET request to retrieve the data as needed.

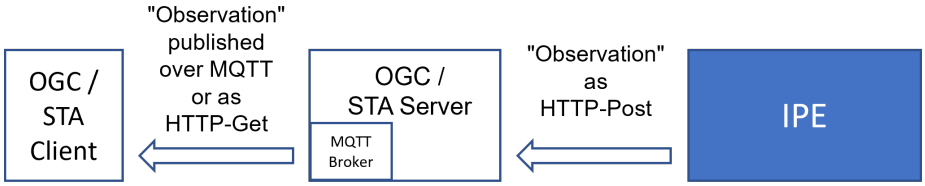


Figure 9: Figure 6.3.1.2-1: Message flow from IPE to OGC/STA Server to OGC/STA Client

### 6.3.2 Configuration of the oneM2M CSE

**6.3.2.0 General Configuration Aspects** The IPE needs to perform configuration steps on the hosting CSE.

The IPE shall register itself as an Application Entity (AE) that is represented as an *<AE>* resource in a oneM2M resource tree.

The CSE uses notifications to communicate new events to the IPE (AE). Therefore, the *<AE>* resource shall have the *requestReachability* (rr) attribute set to 'true'.

The *<AE>* resource shall have a *pointOfAccess* (poa) attribute giving the protocol and address that the IPE is going to use to receive notifications from the CSE.

The message flow for the creation of an *<AE>* resource is shown in Figure 6.3.2.0-1: 1) The IPE requests to register an *<AE>* resource on the hosting CSE. 2) The hosting CSE evaluates the request, performs the appropriate checks, and registers the *<AE>* resource. 3) The hosting CSE responds with a successful result response upon successful creation of the *<AE>* resource; otherwise, it responds with an error.

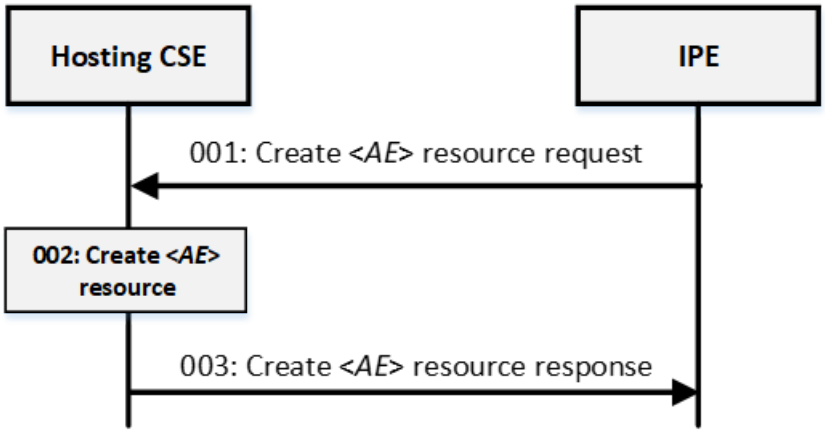


Figure 10: Figure 6.3.2.0-1: Message flow of an *<AE>* resource creation in oneM2M

**6.3.2.1 Communication direction oneM2M CSE towards IPE** It needs two *<container>* resources in the CSE for the operation of the IPE, one for outgoing data and one for incoming data. The *<container>* resource that is appointed to hold the data to be forwarded to the OGC/STA side (outgoing data) has to be created, if not already existing.

The message flow for the creation of a *<container>* resource is shown in Figure 6.3.2.1-1: 1) The IPE sends a request to create a *<container>* resource. 2) The hosting CSE evaluates the request, performs the appropriate checks, and creates the *<container>* resource. 3) The hosting CSE responds with a successful result response upon successful creation of the *<container>* resource; otherwise, it responds with an error.

A *<subscription>* resource shall be created under this *<container>* resource.

The *<subscription>* resource shall have the *notificationURI* attribute set to the *resourceID* of the *<AE>* resource.

The message flow for the creation of an *<subscription>* resource is shown in Figure 6.3.2.1-2: 1) The IPE sends a creation request for a *<subscription>*

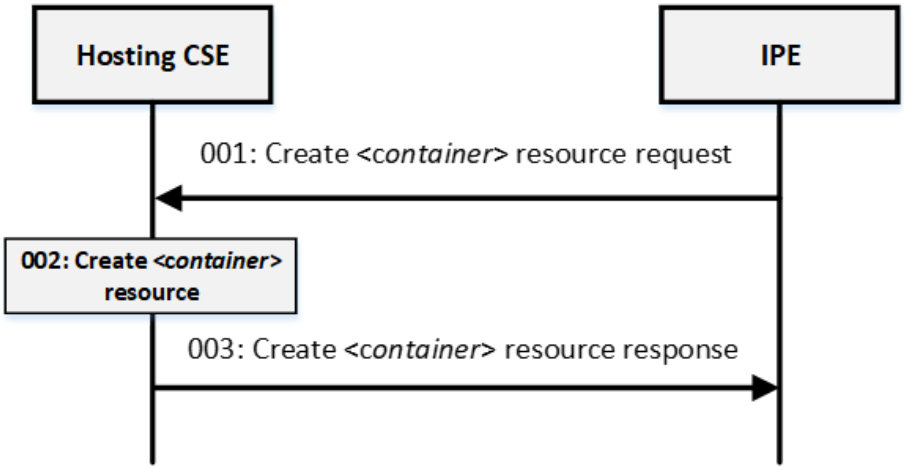


Figure 11: Figure 6.3.2.1-1: Message flow of an *<container>* resource creation in oneM2M

resource to the *<container>* resource that is appointed to hold the data to be forwarded to the OGC/STA side. 2) The hosting CSE evaluates the request and performs the appropriate checks and creates the *<subscription>* resource. 3) The hosting CSE responds with a successful result response of the *<subscription>* resource creation; otherwise, it responds with an error.

The CSE is now prepared to send data from oneM2M to OGC / STA via the IPE. As shown in Figure 6.3.2.1-3, a oneM2M Application Entity (AE), triggered by a sensor, sends data to the CSE by creating a *<contentInstance>* resource under the *<container>* resource that was appointed for outgoing data. Since the IPE has subscribed to this *<container>* resource it receives a notification message along with all attributes of the *<contentInstance>* resource when new data arrives. The IPE maps the data from oneM2M to OGC / STA as described in 6.1 .

**6.3.2.2 Communication direction IPE towards oneM2M CSE** The *<container>* resource that is appointed to hold the data from the OGC/STA side (incoming data) has to be created, if not already existing. The message flow for the creation of a *<container>* resource is shown in Figure 6.3.2.1-1.

The CSE is now prepared to receive data from OGC / STA via the IPE. The IPE sends data as *<contentInstance>* resources to the dedicated *<container>* resource. If other oneM2M Application Entities are interested in this data, they may subscribe to the dedicated *<container>* resource. Alternatively, they can retrieve *<contentInstance>* resources from it in polling mode.

In Figure 6.3.2.2-1, the IPE (AE) sends data as *<contentInstance>* resources to the dedicated *<container>* resource. Subsequently, the AE receives a notifi-

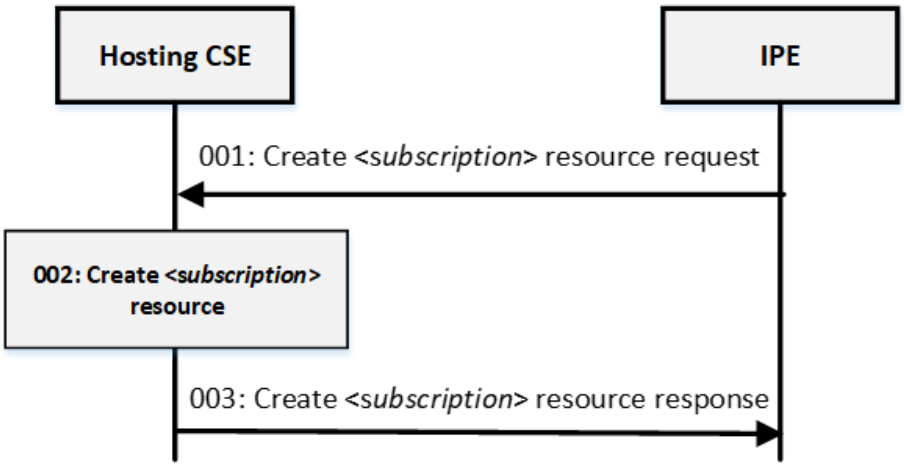


Figure 12: Figure 6.3.2.1-2: Message flow of an *<subscription>* resource creation in oneM2M

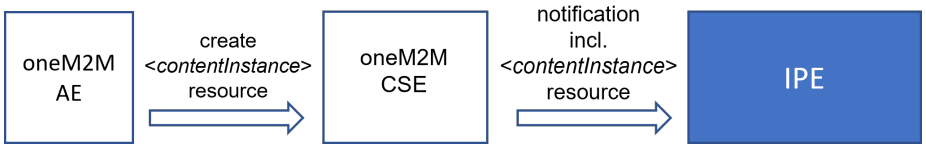


Figure 13: Figure 6.3.2.1-3: Message flow from AE to CSE to IPE

cation along with data contained in a *<contentInstance>* resource every time when the IPE creates new data.

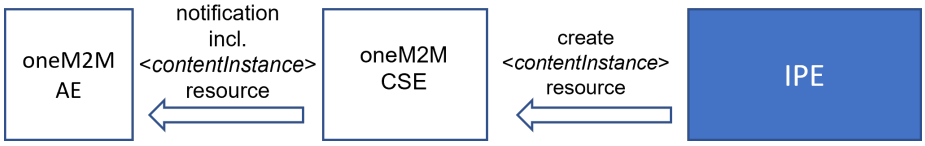


Figure 14: Figure 6.3.2.2-1: Data message flow from IPE to CSE to AE

# History

This clause shall be the last one in the document and list the main phases (all additional information will be removed at the publication stage).

| Publication history | Publication history | Publication history |
|---|---|---|
| V1.x.x | <yyyy-mm-dd > | <Milestone> |

| Version (to be removed on publication) | Date (to be removed on publication) | Draft history (to be removed on publication) |
|---|---|---|
| V5.0.0 | 2024-03-01 | Includes the following contributions agreed during SDS#58 meeting: SDS-2023-0219R01-initial_OGC_intro |
| V5.1.0 | 2024-09-13 | Includes the following contributions agreed during SDS#66 meeting: SDS-2024-0064R02_architecture_model and editorials agreed during SD#S66 |

| Version (to be removed on publication) | Date (to be removed on publication) | Draft history (to be removed on publication) |
|---|---|---|
| V5.2.0 | 2025-03-27 | Includes the following contributions agreed during SDS#68 meeting: SDS-2024-0141R02-ogc_ipe_communication_schema and SDS-2025-0016R02-ogc_ipe_configuration_aspects agreed during SDS#68 |
| V5.3.0 | 2025-04-04 | Includes the following contributions agreed during SDS#69 meeting: SDS-2025-0017R04 ogc_ipe_configuration_aspects_supl agreed during SDS#69, Editorials: followed the convention to write Attributes in italics. |
| V5.4.0 | 2025-08-04 | Includes the following contributions agreed during SDS#70 meeting: SDS-2025-0080R01-text_formatting_updates, Editorials: Supporting information from the template that was intended to assist contributors has been removed to prepare for the transition of the document to change control. Additionally, the abstract and scope have been added compared to v5.3.0. The copyright year has also been updated from 2024 to 2025. |