oneM2M Technical Specification	oneM2M Technical Specification
Document Number	TS-0041-V-5.2.0
Document Name:	oneM2M-SensorThings Interworking\
Date:	2025-03-27
Abstract:	< An abstract of the specification and information that may be used in subsequent electronic searches>
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

Contents

1 Scope

The present document ...

```
EXAMPLE:    The present document provides the necessary adaptions
to the endorsed document.
```

The Scope **shall not** contain requirements.

2 References

The following text block applies.

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

Clause 2.2 shall only contain informative references which are cited in the document itself.

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.

- Use the **EX** style, add the letter "i" (for informative) before the number (which shall be in square brackets) and separate this from the title with a tab (you may use sequence fields for automatically numbering references).
- [i.1] oneM2M Drafting Rules (http://www.onem2m.org/images/files/oneM2M-Drafting-Rules.pdf)

3 Definition of terms, symbols and abbreviations

Delete from the above heading the word(s) which is/are not applicable.

3.1 Terms

Clause numbering depends on applicability.

- A definition shall not take the form of, or contain, a requirement.
- The form of a definition shall be such that it can replace the term in context. Additional information shall be given only in the form of examples or notes (see below).
- The terms and definitions shall be presented in alphabetical order.

For the purposes of the present document, the [following] terms and definitions [given in ... and the following] apply:

Definition format

<defined term>: <definition>

If a definition is taken from an external source, use the format below where [N] identifies the external document which must be listed in Section 2 References.

<defined term>[N]: <definition>

example 1: text used to clarify abstract rules by applying them literally

NOTE: This may contain additional information.

3.2 Symbols

Clause numbering depends on applicability.

For the purposes of the present document, the [following] symbols [given in ... and the following] apply:

Symbol format

<symbol> <Explanation>
<2nd symbol> <2nd Explanation>
<3rd symbol> <3rd Explanation>

3.3 Abbreviations

Abbreviations should be ordered alphabetically.

Clause numbering depends on applicability.

For the purposes of the present document, the [following] abbreviations [given in ... and the following] apply:

Abbreviation format

<ABREVIATION1> <Explanation>
<ABREVIATION2> <Explanation>
<ABREVIATION3> <Explanation>

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.

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

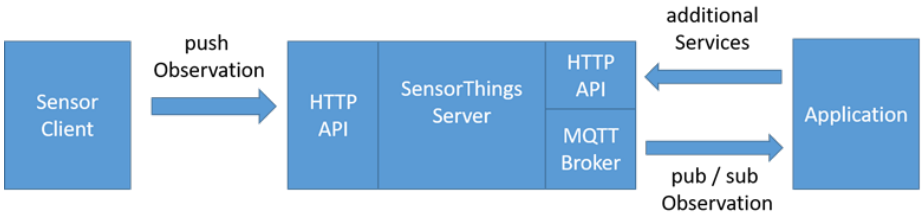

Figure 1: Figure 5-1 STA message flow

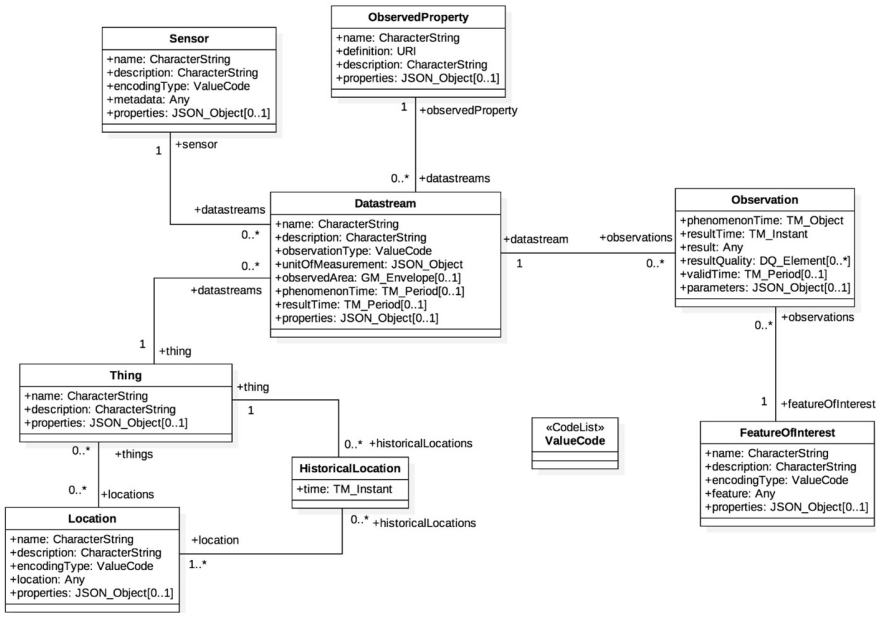

Figure 2: Figure 5-2 STA Sensing Entities Data Model

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].

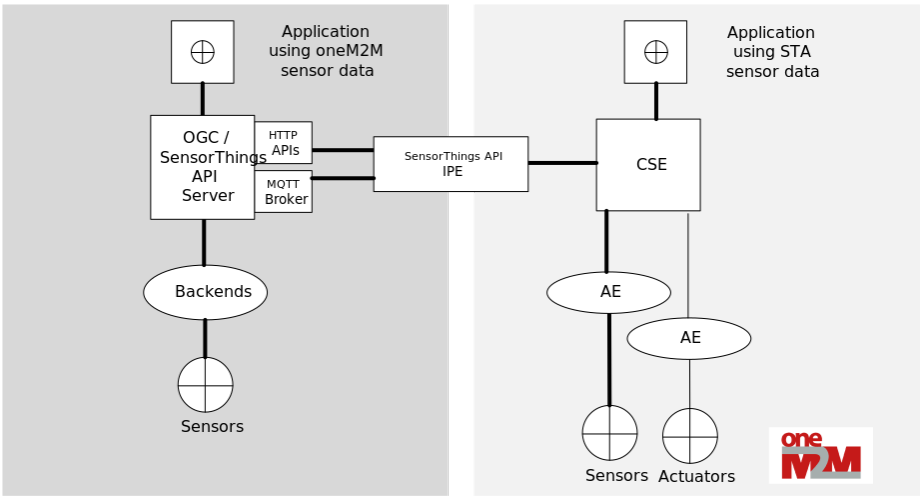

Figure 3: Figure 6.0-1: IPE architecture overview

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>`, 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>` 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.

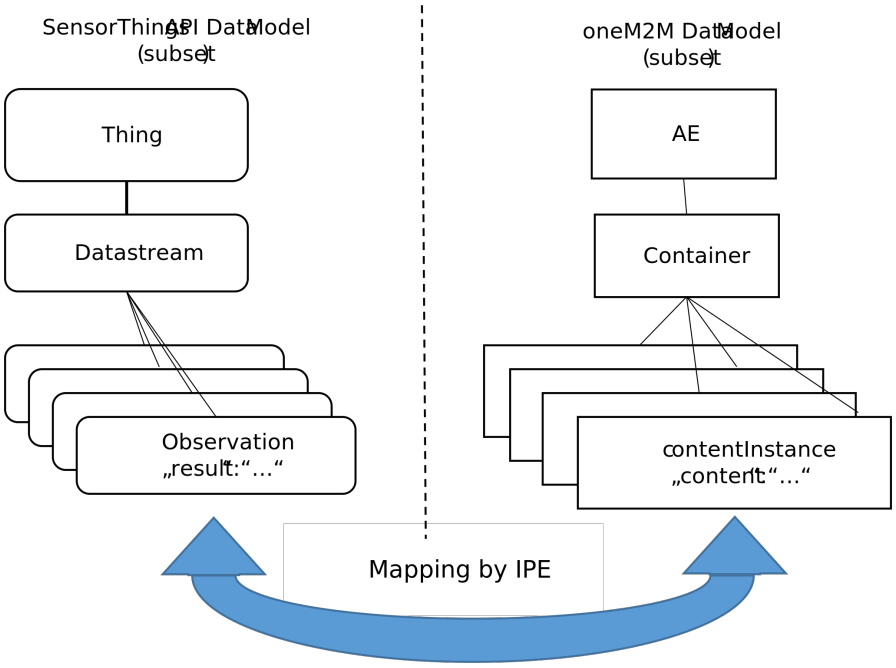

Figure 4: Figure 6.1-1: OGC / STA-to-oneM2M data model mapping

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> to the <container> resource in the CSE containing the desired data. Triggered by a sensor event a new <contentInstance> is added to the <container> by the <AE>. The IPE gets a <notification> containing the <contentInstance> resource. The IPE constructs an "Observation" creation request and copies the 'content' attribute of the <contentInstance> 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.

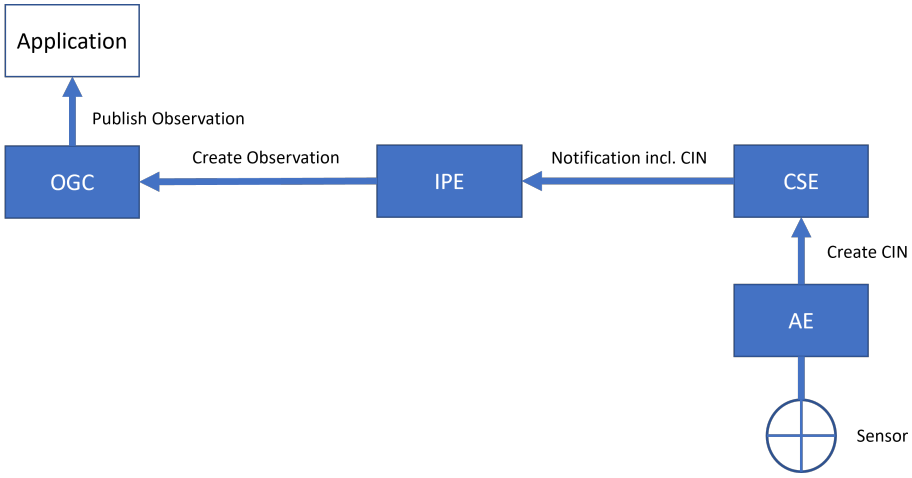

Figure 5: Figure 6.2-1: Communication oneM2M-to-OGC/STA direction

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> 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>. The oneM2M applcation gets the sensor data either by polling the CSE or subscribing to the desired <container> at the CSE.

The following text is to be used when appropriate:

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> and vice versa. This specification defines a 1-to-1 relationship in each direc-

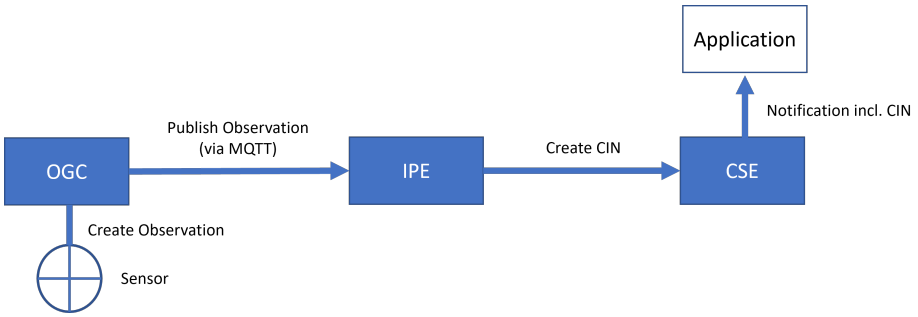

Figure 6: Figure 6.2-2: Communication OGC/STA-to-oneM2M direction

tion between the "Datastream" associated with the "Observation" and the `<container>` associated with the `<contentInstance>`. 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".

6.3.1.2 Communication direction IPE towards OGC/STA Server The IPE requires a destination-"Datastream" to send an "Observation" containing

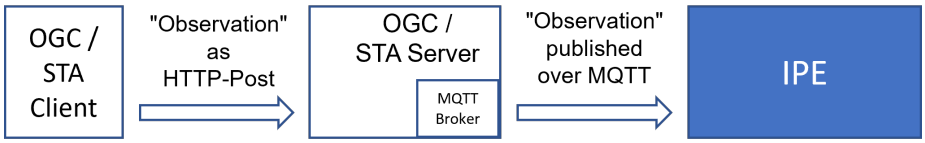

Figure 8: Figure 6.3.1.1-1: Message flow from OGC/STA Client to OGC/STA Server to IPE

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.

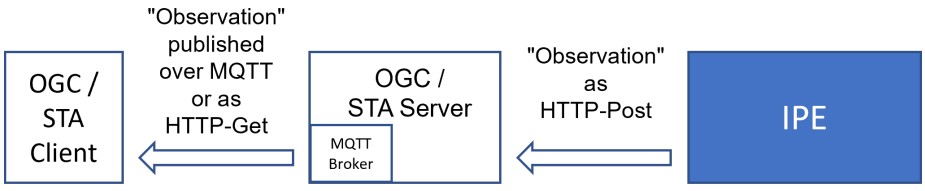

Figure 9: Figure 6.3.1.2-1: Message flow from IPE to OGC/STA Server to OGC/STA Client

Proforma copyright release text block

This text box shall immediately follow after the heading of an element (i.e. clause or annex) containing a proforma or template which is intended to be copied by the user. Such an element shall always start on a new page.

Notwithstanding the provisions of the copyright clause related to the text of the present document, oneM2M grants that users of the present document may freely reproduce the <proformatype> proforma in this {clause|annex} so that it can be used for its intended purposes and may further publish the completed <proformatype>.

<PAGE BREAK>

Annexes

Each annex shall start on a new page (insert a page break between annexes A and B, annexes B and C, etc.).

Use the Heading 9 style for the title and the Normal style for the text.

Annex <A> (Informative/Normative):Remove Informative or Normative as appropriat Title of annex (style H9)

<Text>
<PAGE BREAK>

Annex <B> (Informative/Normative):Remove Informative or Normative as appropriat Title of annex (style H9)

<Text>

B.1 First clause of the annex (style H1)

<Text>

B.1.1 First subdivided clause of the annex (style H2)

<Text>

<PAGE BREAK>

The following text is to be used when appropriate:

Annex <y>:

Bibliography The annex entitled “Bibliography” is optional.

It shall contain a list of standards, books, articles, or other sources on a particular subject which are not mentioned in the document itself

It shall not include references mentioned in the document.

Use the Heading 9 style for the title and B1+ or Normal for the text.

- <Publication>: “<Title>”.

OR

<Publication>: “<Title>”.

<PAGE BREAK>

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

Version (to be removed on publication)	Date (to be removed on publication)	Draft history (to be removed on publication)
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
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