

ETSI TS 118 141 V5.4.1 (2025-11)



**oneM2M;
SensorThings Interworking
(oneM2M TS-0041 version 5.4.1 Release 5)**



Reference

DTS/oneM2M-0041r05v1

Keywords

API, IoT, M2M, sensor

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from the
[ETSI Search & Browse Standards](#) application.

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on [ETSI deliver](#) repository.

Users should be aware that the present document may be revised or have its status changed,
this information is available in the [Milestones listing](#).

If you find errors in the present document, please send your comments to
the relevant service listed under [Committee Support Staff](#).

If you find a security vulnerability in the present document, please report it through our
[Coordinated Vulnerability Disclosure \(CVD\)](#) program.

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/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.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2025.
All rights reserved.

Contents

Foreword.....	4
Modal verbs terminology.....	4
1 Scope	5
2 References	5
2.1 Normative references	5
2.2 Informative references.....	5
3 Definition of terms, symbols and abbreviations.....	5
3.1 Terms.....	5
3.2 Symbols.....	5
3.3 Abbreviations	5
4 Conventions.....	6
5 Introduction to OGC SensorThings API	6
6 Architecture model of OGC/STA and oneM2M interworking	8
6.0 Overview	8
6.1 OGC/STA-to-oneM2M data model mapping.....	8
6.2 Communication flow	9
6.3 Configuration aspects.....	10
6.3.0 Introduction.....	10
6.3.1 Configuration of OGC/STA server side.....	11
6.3.1.0 Overview	11
6.3.1.1 Communication direction OGC/STA server towards IPE.....	11
6.3.1.2 Communication direction IPE towards OGC/STA server.....	11
6.3.2 Configuration of the oneM2M CSE.....	12
6.3.2.0 General configuration aspects	12
6.3.2.1 Communication direction oneM2M CSE towards IPE	12
6.3.2.2 Communication direction IPE towards oneM2M CSE	14
History	15

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the [ETSI IPR online database](#).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Specification (TS) has been produced by ETSI Partnership Project oneM2M (oneM2M).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

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

2.1 Normative references

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

Referenced documents which are not found to be publicly available in the expected location might be found in the [ETSI docbox](#).

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

- [1] Open Geospatial Consortium (OGC): "[OGC SensorThings API Part 1: Sensing Version 1.1](#)".
- [2] [ETSI TS 118 133](#): "oneM2M; Interworking Framework".
- [3] [ETSI TS 118 101](#): "oneM2M; Functional Architecture".

2.2 Informative references

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

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

- [i.1] [oneM2M Drafting Rules](#).

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

Void.

4 Conventions

The keywords "Shall", "Shall not", "May", "Need not", "Should", "Should not" in the present 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 divided in two parts. SensorThings API Part 1 [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.

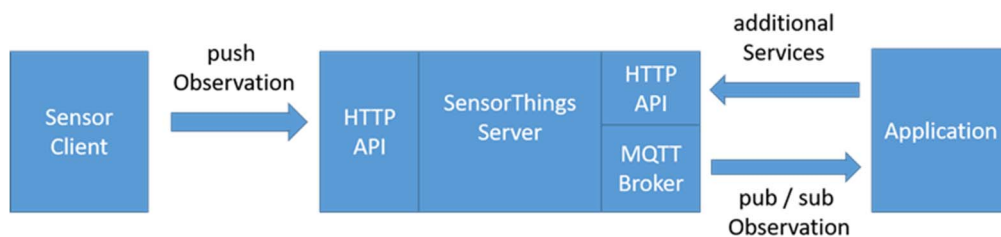


Figure 5-1: STA message flow

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

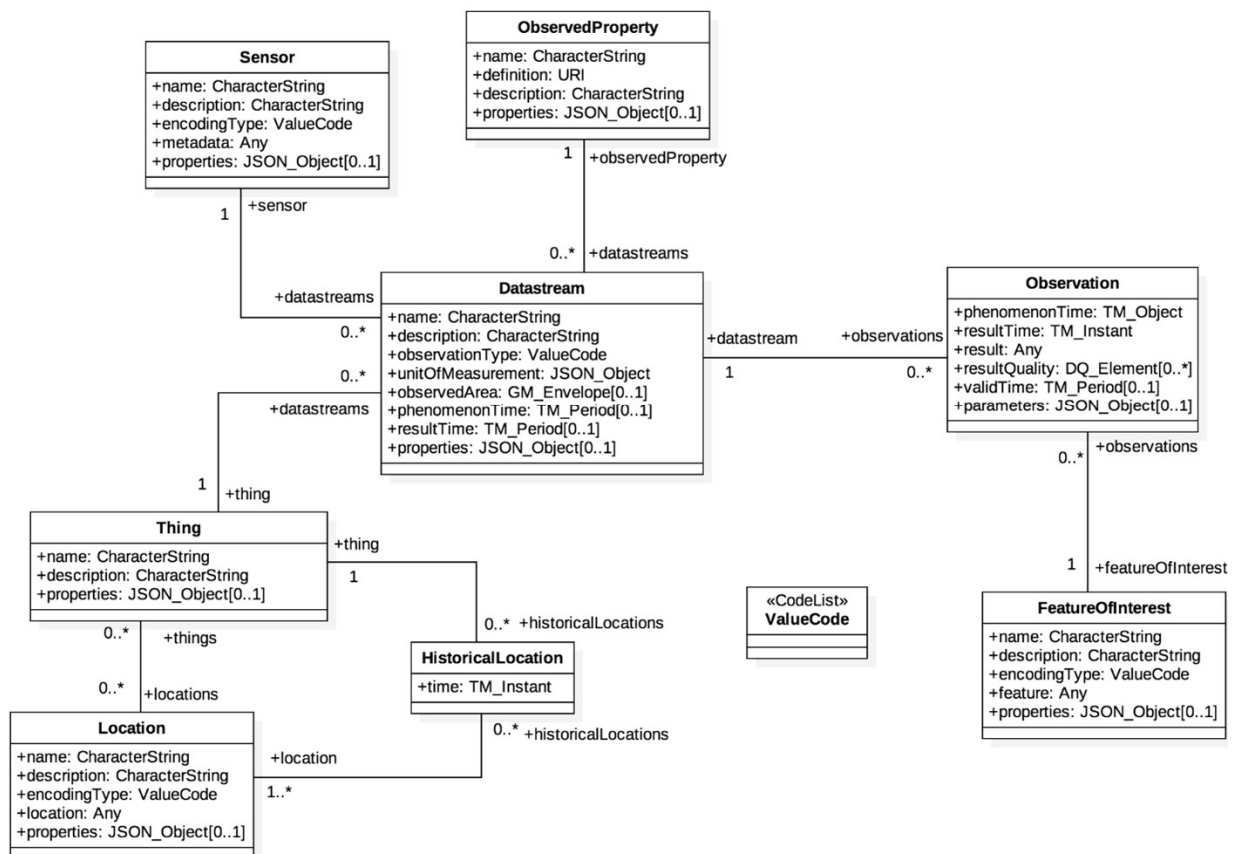


Figure 5-2: Sensing Entities data model

In the Sensing Entities data model events or sensor data are called Observation entities. Before a sensor is able to push an observation to the server it needs at least a Thing entity and a Datastream entity. These have to be created beforehand. One Thing entity might have different Sensor entities, one Location entity or many HistoricalLocations entities.

The Sensing Entities data model and the purpose of data within the data model discloses mainly two data characteristics associated with a Thing entity:

- data observations originated by sensors or commands sent to interact with actuators can be seen as IoT data from oneM2M point of view;
- data embedded in the Sensing Entities data model, like HistoricalLocation entities, can be seen as data for documentation purposes.

6 Architecture model of OGC/STA and 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 a 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 a 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 [1].

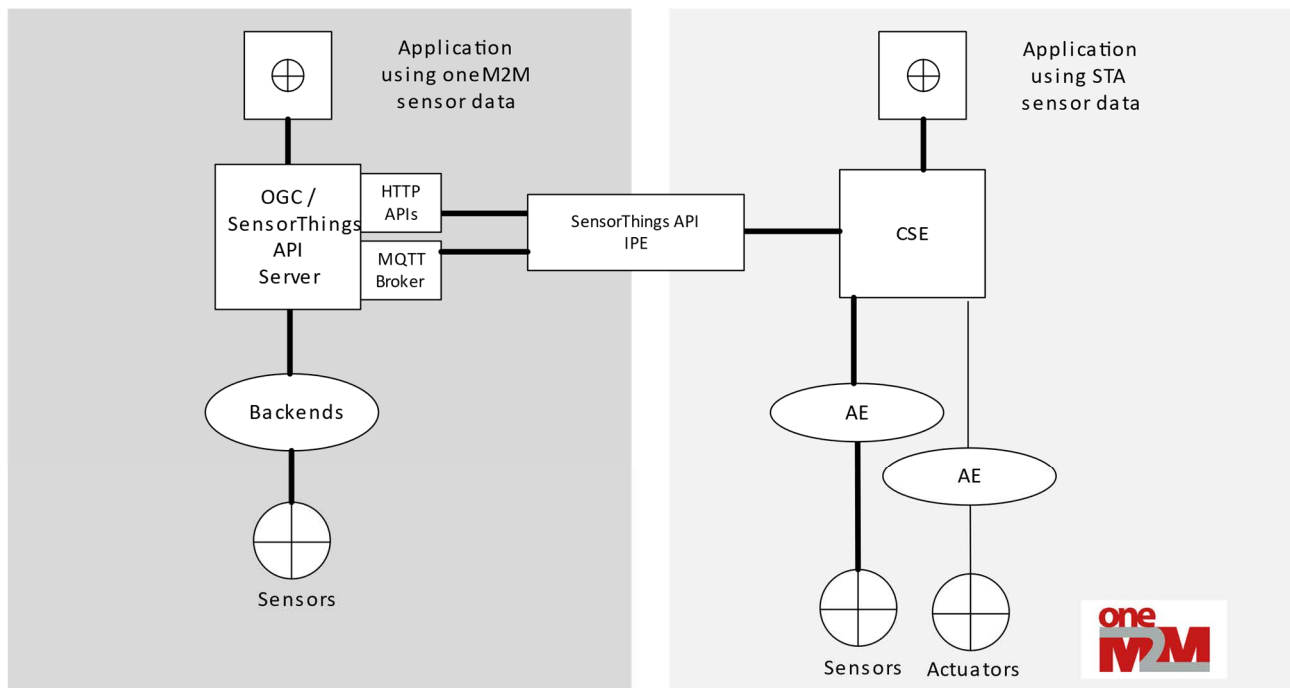


Figure 6.0-1: IPE architecture overview

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 Sensing Entities data model has to be represented in the hosting CSE. The Sensing Entities 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 product IDs).

The IPE shall map the *result* property of an OGC/STA Observation entity 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* property of an Observation entity is according to SensorThings API [1] 'any' and depends on the *observationType* property defined in the associated Datastream entity. The *content* attribute of a oneM2M *<contentInstance>* resource 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 entity as *phenomenonTime* property and in the *<contentInstance>* resource as *creationTime* attribute, 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* property or the *content* attribute.

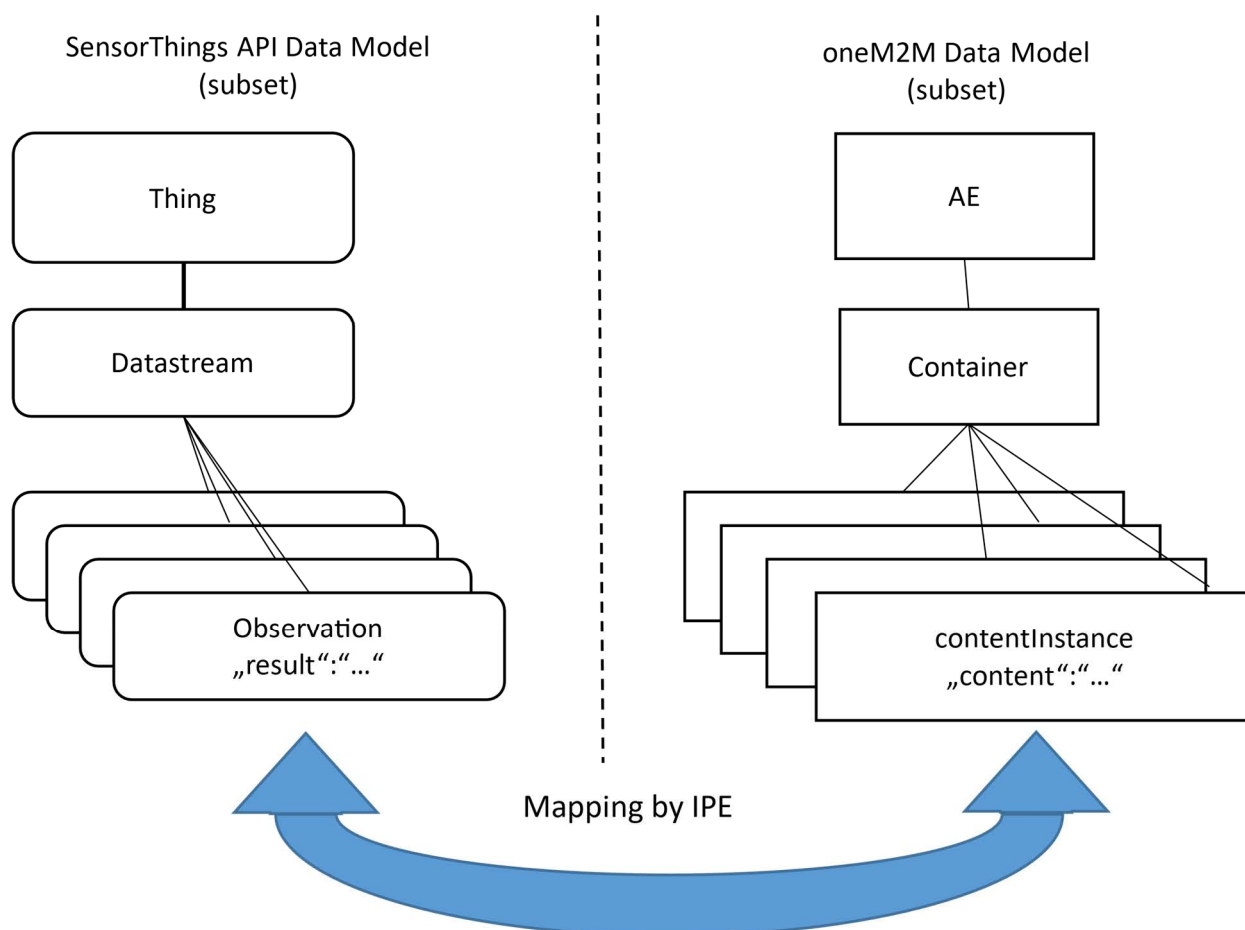


Figure 6.1-1: OGC/STA-to-oneM2M data model mapping

6.2 Communication flow

Figure 6.2-1 shows the oneM2M-to-OGC/STA direction of the communication flow. 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 containing the desired data. Triggered by a sensor event a new *<contentInstance>* resource is added to the *<container>* resource by the AE. The IPE gets a notification containing the *<contentInstance>* resource. The IPE constructs an Observation entity creation request and copies the *content* attribute of the *<contentInstance>* resource to the *result* property of the Observation entity and sends the request to a Datastream entity to be created as detailed in clause 6.3.1 at the OGC/STA server. The OGC/STA application gets the sensor data either by polling the OGC/STA server or subscribing to the corresponding Datastream entity at the MQTT broker of the OGC/STA server.

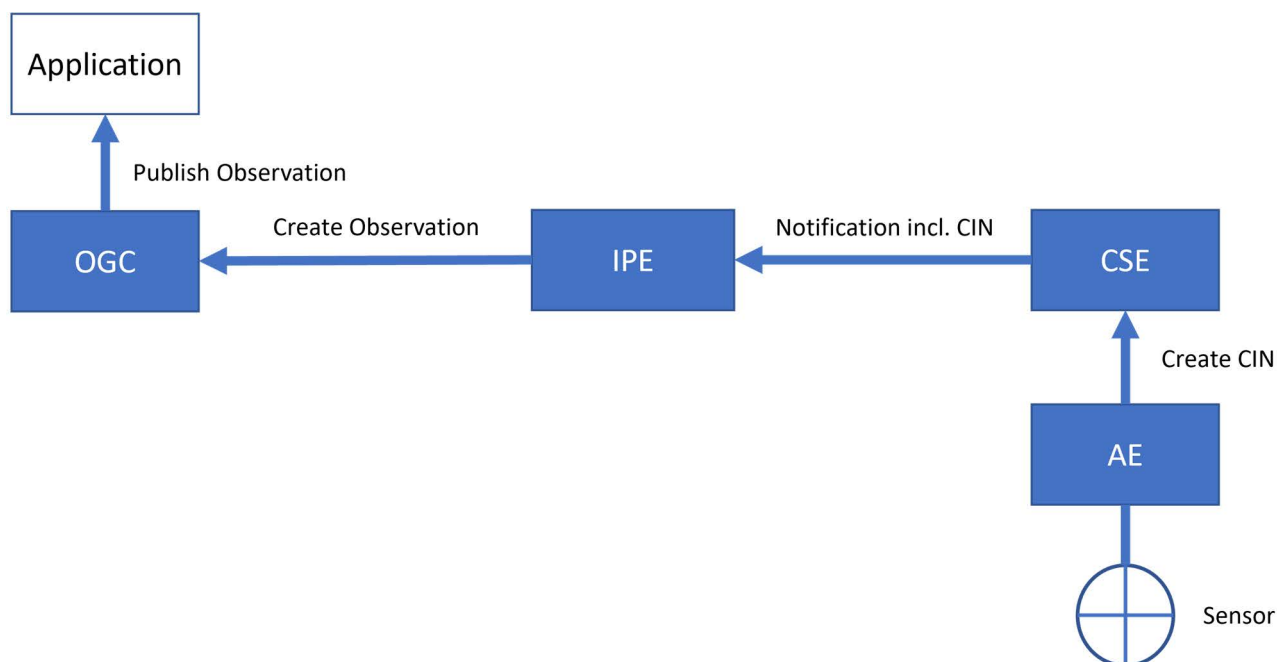


Figure 6.2-1: Communication flow in oneM2M-to-OGC/STA direction

Figure 6.2-2 shows the OGC/STA-to-oneM2M direction of the communication flow. The IPE subscribes to the desired Datastream entity of the MQTT broker at the OGC/STA server. The OGC/STA server publishes a new Observation entity via the MQTT broker triggered by an OGC/STA sensor. The IPE creates a *<contentInstance>* resource in a *<container>* resource to be created as detailed in clause 6.3.2 in the CSE and copies the *result* property of the Observation entity to the *content* attribute of the *<contentInstance>* resource. The oneM2M application gets the sensor data either by polling the CSE or subscribing to the desired *<container>* resource at the CSE.

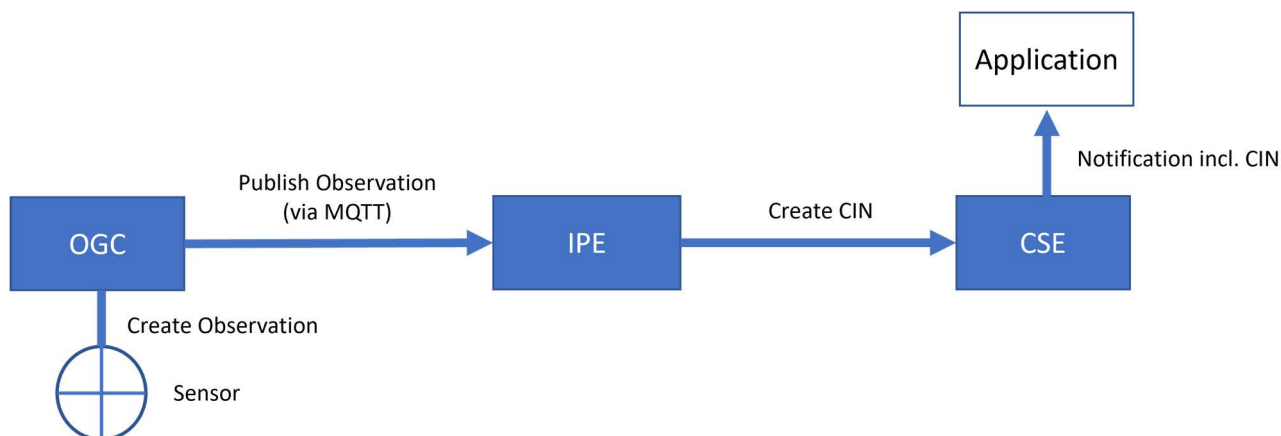


Figure 6.2-2: Communication flow in OGC/STA-to-oneM2M direction

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 clause 6.0, the IPE maps data from an OGC/STA Observation entity to a oneM2M *<contentInstance>* resource and vice versa. The present document defines a 1-to-1 relationship in each direction between the Datastream entity associated with the Observation entity and the *<container>* resource associated with the *<contentInstance>* resource. An IPE may implement multiple 1-to-1 relationships.



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 entity according to the Sensing Entities data model [1] belongs to a Datastream entity (see figure 5-2). The IPE shall subscribe to the Datastream entity 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/Datastream(8715)}. Upon successful subscription, the IPE will receive every Observation entity pushed to that Datastream entity.

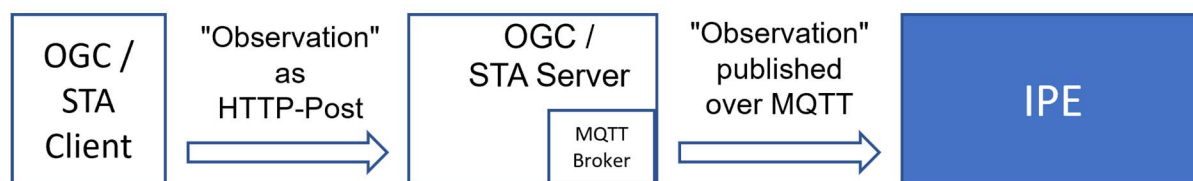


Figure 6.3.1.1-1: Message flow from OGC/STA client to OGC/STA server to IPE

6.3.1.2 Communication direction IPE towards OGC/STA server

The IPE requires a destination Datastream entity to send an Observation entity containing data from the oneM2M side. If no associated Datastream entity 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 entity 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 entity with a Datastream entity and shall be available during IPE operation. In addition to the Datastream entity, other entities of the Sensing Entities dataModel [1], such as Location entity or Sensor entity, may be created.

The creation of entities like Datastream entity and Thing entity requires several mandatory properties that shall be known at configuration time (e.g. the *name* property and the *description* property). These properties may be automatically derived, for example, from the *label* attribute or the *resourceName* attribute 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 [1].

Once the destination Datastream entity is created, the IPE can send an Observation entity to the OGC/STA server as HTTP POST message. An interested OGC/STA client can subscribe to the destination Datastream entity at the MQTT broker of the OGC/STA server to receive each Observation entity 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.

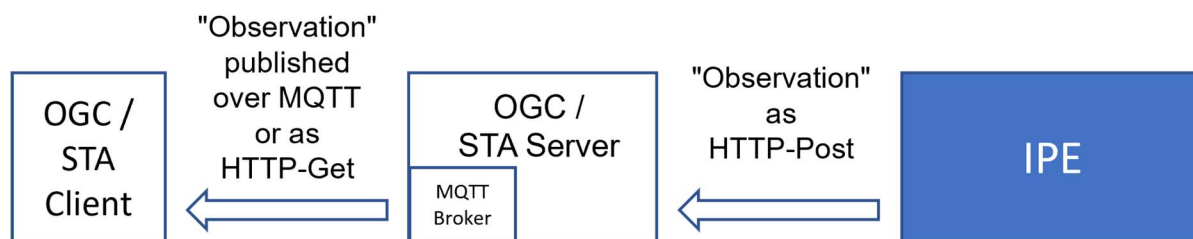


Figure 6.3.1.2-1: Message flow from IPE to OGC/STA server to OGC/STA client

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 $\langle AE \rangle$ resource in a oneM2M resource tree.

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

The $\langle AE \rangle$ 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 $\langle AE \rangle$ resource is shown in figure 6.3.2.0-1.

- 1) The IPE requests to register an $\langle AE \rangle$ resource on the hosting CSE.
- 2) The hosting CSE evaluates the request, performs the appropriate checks, and registers the $\langle AE \rangle$ resource.
- 3) The hosting CSE responds with a successful result response upon successful creation of the $\langle AE \rangle$ resource. Otherwise, it responds with an error.

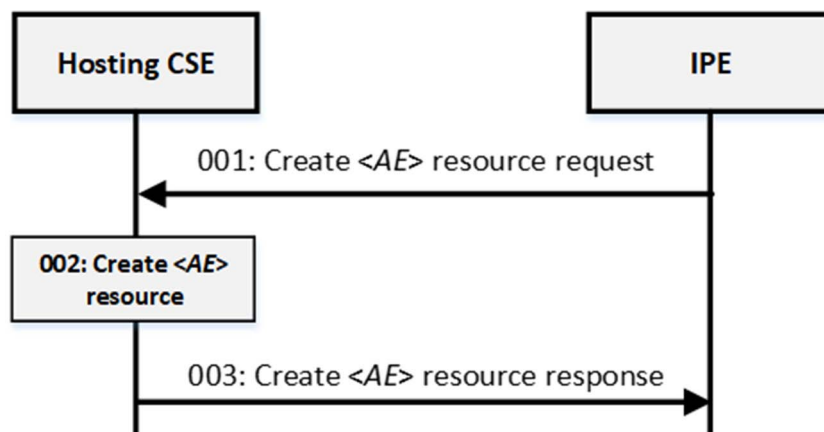


Figure 6.3.2.0-1: Message flow of an $\langle AE \rangle$ resource creation in oneM2M

6.3.2.1 Communication direction oneM2M CSE towards IPE

Two $\langle container \rangle$ resources are required in the CSE for the operation of the IPE, one for outgoing data and one for incoming data. The $\langle container \rangle$ 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 $\langle container \rangle$ 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.

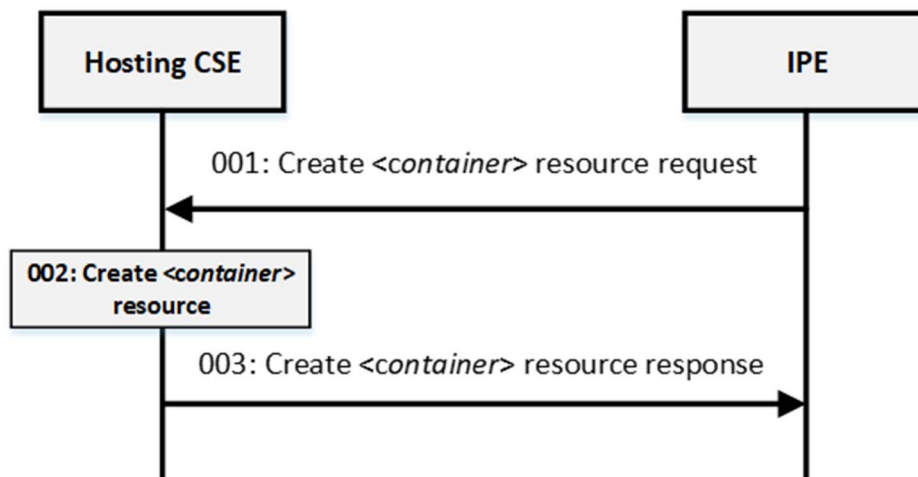


Figure 6.3.2.1-1: Message flow of a *<container>* resource creation in oneM2M

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

The *<subscription>* resource shall have the *notificationURI* attribute set to the Resource ID 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>* 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 upon the successful creation of the *<subscription>* resource. Otherwise, it responds with an error.

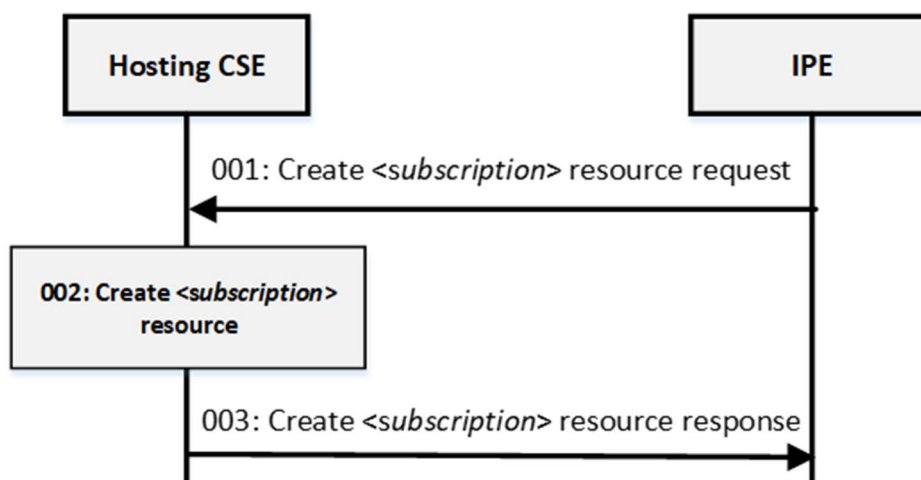


Figure 6.3.2.1-2: Message flow of a *<subscription>* resource creation in oneM2M

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 clause 6.1.



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

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



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

History

Document history		
V5.4.1	November 2025	Publication