ETSI GS MEC 046 V3.2.1 (2025-08)

GROUP SPECIFICATION

Multi-access Edge Computing (MEC); Sensor-sharing API

Disclaimer

The present document has been produced and approved by the Multi-access Edge Computing (MEC) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
RGS/MEC-0046v321SensorSharing

Keywords
API, MEC, 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

Contents

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 Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Multi-access Edge Computing (MEC).

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 focuses on a MEC Sensor-sharing Service that facilitates the provision of sensors' information and data in a multi-access environment. It describes the sensors information flows, required information and operations including authorization and access control. The present document specifies the necessary API with the data model and data format.

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] ETSI GS MEC 003: "Mobile Edge Computing (MEC); Framework and Reference Architecture".

[2] ETSI TS 118 101 (V3.22.0): "oneM2M; Functional Architecture (oneM2M TS-0001 version 3.22.0 Release 3)".

[3] ETSI GS MEC 009: "Mobile Edge Computing (MEC); General principles for Mobile Edge Service APIs".

[4] ETSI TS 103 264: "SmartM2M; Smart Applications; Reference Ontology and oneM2M mapping".

[5] ETSI TS 118 112 (V3.7.3): "oneM2M; Base Ontology (oneM2M TS-0012 version 3.7.3 Release 3)".

[6] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".

[7] IETF RFC 8446: "The Transport Layer Security (TLS) Protocol Version 1.3".

[8] ETSI TS 133 210: "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; Network Domain Security (NDS); IP network layer security (3GPP TS 33.210)".

[9] IETF RFC 6749: "The OAuth 2.0 Authorization Framework".

[10] IETF RFC 6750: "The OAuth 2.0 Authorization Framework: Bearer Token Usage".

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.

[i.2] Ontology of units of Measures (OM).

[i.3] W3C® XSD Data types.

[i.4] ETSI GR MEC 001: "Multi-access Edge Computing (MEC); Terminology".

[i.5] OpenAPI™ Specification.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GR MEC 001 [i.4] and in ETSI TS 103 264 [4] apply.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GR MEC 001 [i.4] and the following apply:

IoT	Internet of Things
M2M	Machine to Machine
OM	Ontology of units of Measure
SAREF	Smart Applications REFerence
W3C®	World Wide Web Consortium

4 Overview

The present document specifies the MEC Sensor-sharing Service that facilitates the provision of sensors' information and data in a multi-access environment.

Clause 5 introduces the description of the MEC Sensor-sharing Service, and it lists the provided functionalities. It also describes the procedures that the MEC Sensor-sharing Service offers.

The information that can be exchanged over the MEC Sensor-sharing Service API is described in clause 6 which provides detailed descriptions of all information elements that are used for this service.

Clause 7 defines the MEC Sensor-sharing Service API providing detailed information of how information elements are mapped into a RESTful API design.

5 Description of the service (informative)

5.1 Introduction

MEC Sensor-sharing service defines how MEC applications can interact with one or more sensors using the Sensor-sharing API. Using the API offered by this service, the MEC applications can perform the active sensors monitoring, manage the devices and retrieve the data collected by the sensors. The Sensor-sharing service is registered and discovered over the Mp1 reference point defined in ETSI GS MEC 003 [1].

The MEC Sensor-sharing API provides both lookup mechanisms - where the information requested is reported only once for each request - and the subscription mechanism - where the information requested could be reported multiple times for each request, following a publish/subscribe approach.

The MEC Sensor-sharing API offers the following services:

- Sensor discovery: a MEC application can request information about the available sensors, the request can ask for sensors with specific features or location.
- Sensor status provision: the Sensor-sharing API can provide the current status of a sensor or of a set of sensors.
- Sensor data provision: a MEC application can use this service for receiving data from one or more sensors.
- Sensor management: a MEC application can remotely configure one or more sensors.

The MEC Sensor-sharing service should implement the required interactions with the sensors to correctly provide to the MEC applications the information and data from the sensors. The definition of which type of sensor the MEC Sensor-sharing service should interact with and how this interaction should be performed are beyond the scope of the present document.

5.2 Relation with oneM2M

Defining the MEC Sensor-sharing API, the alignment of the present document with other efforts already defined in the IoT domain has been kept in strong consideration. Specifically. The oneM2M standards [i.1] and [2] have been analysed as one reference solution for the interoperability of IoT devices.

During the definition of MEC Sensor-sharing API services, the list of services defined by oneM2M [2] has been considered, to be sure to cover all the needed functionalities and to have a clear mapping among the two standards. Specifically, the oneM2M services considered relevant also for MEC 046 are Discovery, Group Management, Subscription & Notification, Device Management, Location and Semantics. All the other services defined by oneM2M have been considered as out of scope for the present document, because not relevant - e.g. the Registration one, which is not relevant because the specification of MEC Sensor-sharing API focuses on the provisioning of information and data from sensors - or already supported by other MEC standards, e.g. the Security part already covered by ETSI GS MEC 009 [3].

5.3 Sequence diagrams

5.3.1 Introduction

Clauses 5.3.2 to 5.3.8 describe how the API Clients interact with the Sensor-sharing Service over Sensor-sharing API to retrieve information about the available sensors and their characteristics, gather sensor data, and manage the sensors' configuration. The Sensor-sharing API supports both queries and subscriptions (pub/sub mechanism) that can be used over RESTful API. The sequence diagrams that are relevant to the Sensor-sharing Service are presented.

5.3.2 Sensor Discovery Lookup

The Sensor Discovery Lookup is the procedure for applications acquiring the sensor(s) currently available and the related characteristics.

The Sensor Discovery Lookup procedure is illustrated in Figure 5.3.2-1.

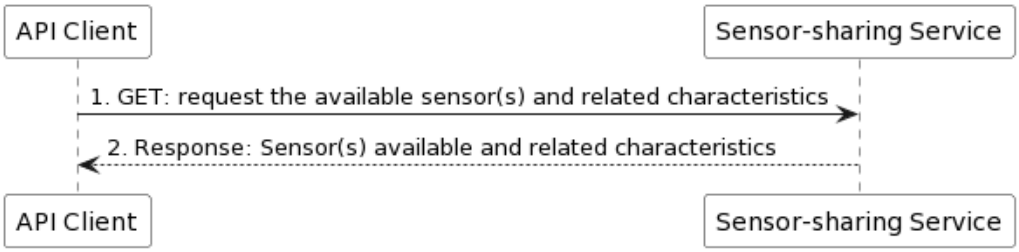

Figure 5.3.2-1: Flow of Sensor Discovery Lookup

1) The MEC application looks up information about the available sensor(s) and related characteristics by sending a request to the resource representing the sensor information. The request may optionally include one or more query parameters specifying the sub-region of interest, or specific sensor characteristics that the MEC application is interested in.

2) The Sensor-sharing Service returns a response with a message body including the information of the sensor(s) according to the query parameters if the sensor discovery lookup is accepted.

5.3.3 Sensor Discovery Subscription

The Sensor Discovery Subscription is the procedure for applications to request to receive notifications about sensors when sensor information changes, e.g. a new sensor becomes available, or an existing sensor becomes unavailable. Such notifications help the MEC application to be updated about the sensor information. In this procedure, the Sensor-sharing Service will continue to report the subscribed information until the subscription is cancelled, or an optional specified time limit is reached.

The Sensor Discovery Subscription procedure is illustrated in Figure 5.3.3-1.

Figure 5.3.3-1: Flow of Sensor Discovery Subscription

1) The MEC application subscribes to Sensor Discovery notifications by requesting the creation of a resource containing the subscription details, which include the sensor(s) identifier, and a callbackURL for receiving the sensor information.

2) The Sensor-sharing Service returns a response with a resource URI containing the subscriptionId.

3) The Sensor-sharing Service reports the up-to-data subscribed information to the MEC application by sending a message with the message body containing the Sensor Discovery notification to the callbackURL, which includes sensor information.

4) The MEC application returns a response with the code 204.

5.3.4 Sensor Status Lookup

The Sensor Status Lookup is the procedure for applications acquiring the status (e.g. active or error state) of sensor(s).

The Sensor Status Lookup procedure is illustrated in Figure 5.3.4-1.

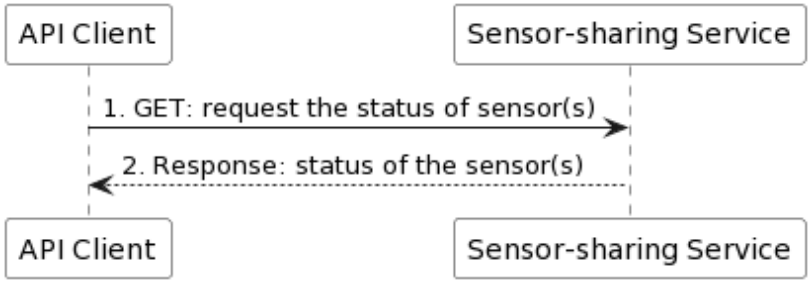

Figure 5.3.4-1: Flow of Sensor Status Lookup

1) The MEC application looks up the current status of sensor(s) by sending a request to the resource representing the sensor(s) status information, which includes the identifier of the sensor(s) of interest.

2) The Sensor-sharing Service returns a response with a message body including the current status of the sensor(s).

5.3.5 Sensor Status Subscription

The Sensor Status Subscription is the procedure for applications to request to receive notifications about sensors when they change their status (e.g. entering an error state). These notifications may help the MEC application to regularly track the sensor(s) status. In this procedure, the Sensor-sharing Service will continue to report the subscribed information until the subscription is cancelled, or an optional specified time limit.

The Sensor Status Subscription procedure is illustrated in Figure 5.3.5-1.

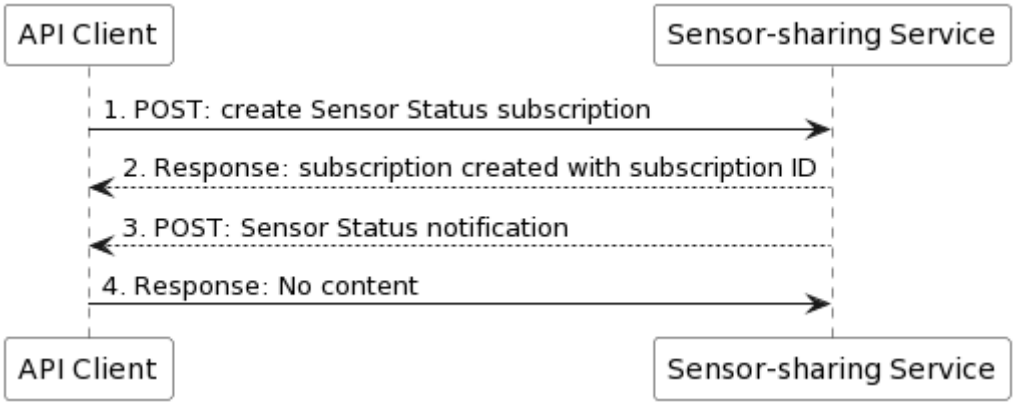

Figure 5.3.5-1: Flow of Sensor Status Subscription

1) The MEC application subscribes to Sensor Status notification by requesting the creation of a resource containing the subscription details, which includes the identifier of the sensor(s), as well as the callbackURL for receiving the sensor(s) status.

2) The Sensor-sharing Service returns a response with resource URI containing the subscriptionId.

3) The Sensor-sharing Service reports the up-to-data subscribed information to the MEC application by sending a message with the message body containing the Sensor Status notification to the callbackURL, which includes the updated sensor(s) status information.

4) The MEC application returns a response with the code 204.

5.3.6 Sensor Data Lookup

The Sensor Data Lookup is the procedure for applications to acquire the last data collected by the sensor(s).

The Sensor Data Lookup procedure is illustrated in Figure 5.3.6-1.

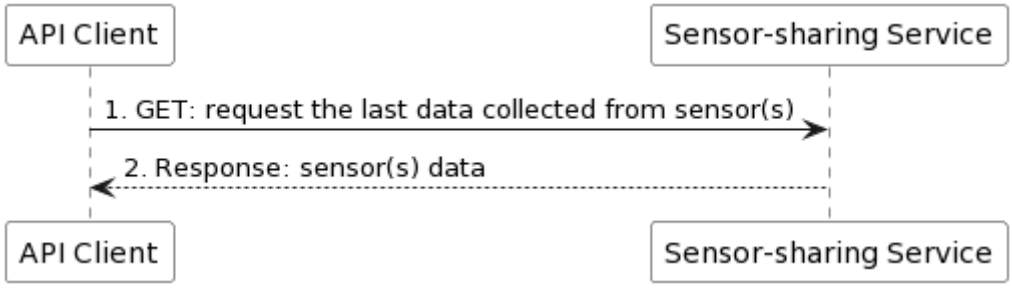

Figure 5.3.6-1: Flow of Sensor Data Lookup

1) The MEC application looks up the last collected data from the sensor(s) by sending a request to the resource representing the sensor data information, which includes the sensor(s) identifier.

2) The Sensor-sharing Service returns a response with a message body including the last data collected by the sensor(s).

5.3.7 Sensor Data Subscription

The Sensor Data Subscription is the procedure for applications to request to receive notifications every time new data are collected from the sensor(s). Such notifications may help the MEC application to regularly receive the data collected by the sensor(s). In this procedure, the Sensor-sharing Service will continue to report the subscribed information until the subscription is cancelled, or an optional specified time limit is reached.

The Sensor Data Subscription procedure is illustrated in Figure 5.3.7-1.

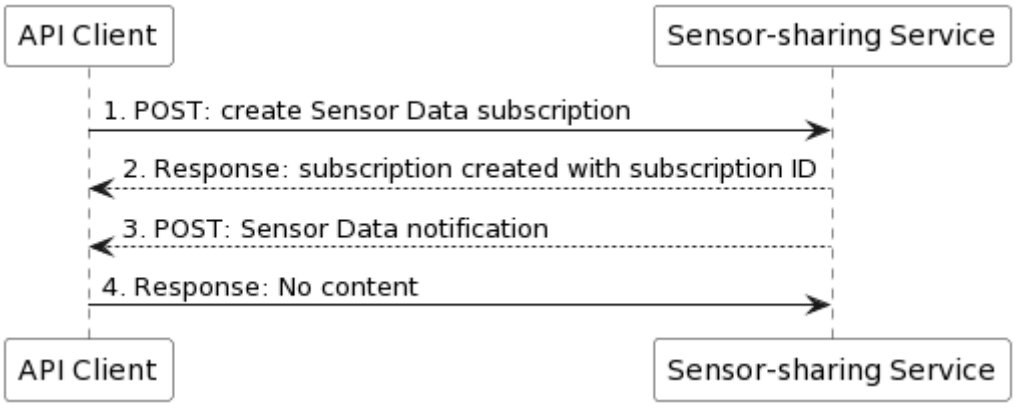

Figure 5.3.7-1: Flow of Sensor Data Subscription

1) The MEC application subscribes to Sensor Data notification by requesting the creation of a resource containing the subscription details, which include the include sensor(s) identifier, as well as the callbackURL for receiving the notifications.

2) The Sensor-sharing Service returns a response with a resource URI containing the subscriptionId.

3) The Sensor-Sharing Service reports the up-to-data subscribed information to the MEC application by sending a message with the message body containing the Sensor Data notification to the callbackURL, which includes the last collected data from the sensor(s).

4) The MEC application returns a response with the code 204.

5.3.8 Sensor Management

The Sensor Management is the procedure for a MEC application to configure specific parameter(s) on the sensor(s), e.g. the sampling rate of a sensor. In this procedure, the MEC application first requires the parameter(s) that can be set on the sensor(s). After the reception of the information about the parameters that can be set, the MEC application provides to the Sensor-sharing Service the value of the parameter(s) to be modified. Lastly, the Sensor-sharing Service provides the response which contains the outcome of the management operation, e.g. the new value of the parameter was set correctly, or the error(s) occurred.

The Sensor Management procedure is illustrated in Figure 5.3.8-1.

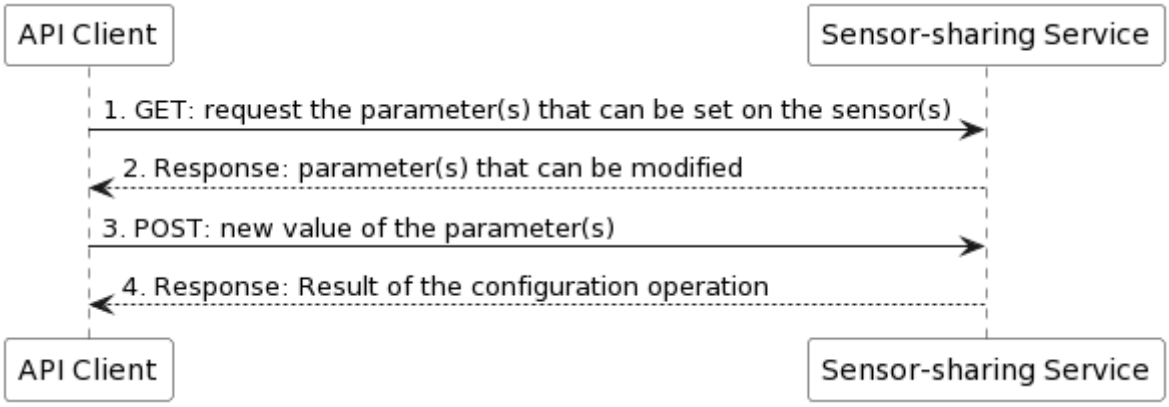

Figure 5.3.8-1: Flow of Sensor Management

1) The MEC application looks up the parameter(s) that can be modified on the sensor(s) by sending a request to the resource containing the sensor(s) management information, which includes the sensor(s) identifier.

2) The Sensor-sharing Service returns a response with a message body including the parameter(s) that can be configured on the sensor(s).

3) The MEC application configures the sensor(s) by sending a message with the message body containing the sensor(s) identifier, as well as the new value of the parameter(s).

4) The Sensor-sharing Service returns a response with code 200 if the new configuration has been set correctly or with code 400 if the configuration was not set. In the last case, the message body contains the details about the error(s).

6 Data model

6.1 Relation with oneM2M and SAREF ontology

Defining the MEC Sensor-sharing data model, the alignment of the present document with other efforts already defined in the IoT domain has been kept in strong consideration. Specifically, for the present document have been analysed, as reference solutions for the semantic interoperability of IoT devices, ETSI TS 103 264 [4], which contains the specification of the Smart Applications REFerence ontology (SAREF) ontology and its extensions [4], clause 4.5. It is important to note that the SAREF specification indicates a clear mapping (see ETSI GS MEC 009 [3], clause 5) with the oneM2M base ontology [5]. Therefore, aligning the data model with SAREF, there will be an alignment also with the oneM2M base ontology.

Specifically, the MEC Sensor-sharing data model will leverage the following SAREF concepts:

- saref:Sensor in ETSI TS 103 264 [4], clause 4.3.2: a device that detects and responds to events or changes in the physical environment such as light, motion, or temperature changes.

- saref:State in ETSI TS 103 264 [4], clause 4.3.5: the state in which a device can be found, e.g. ON/OFF/STANDBY, or ONLINE/OFFLINE. A list of states that are relevant for the purpose of SAREF is proposed, but this list can be extended.
- saref:Property in ETSI TS 103 264 [4], clause 4.3.8: anything that can be sensed, measured or controlled in households, common public buildings or offices. SAREF defines a list of possible properties that can be extended.
- saref:UnitOfMeasure in ETSI TS 103 264 [4], clause 4.3.8: it is a standard for measurement of a quantity, such as a Property. SAREF definition is based on the definition of unit of measure in the Ontology of units of Measure (OM) [i.2]. The Sensor-sharing data model uses as a reference the list of units of measure defined by SAREF and its extensions, but, if needed, this list can be extended, also using some other ontologies rather than the Ontology of units of Measure (OM).

6.2 Resource data types

6.2.1 Type: SensorDiscoveryInfo

This type represents the information about type, properties and characteristics that are related to a sensor.

The attributes of the SensorDiscoveryInfo shall follow the notations provided in Table 6.2.1-1.

Table 6.2.1-1: Definition of type SensorDiscoveryInfo

Attribute name	Data type	Cardinality	Description
sensorIdentifier	String	1	Unique identifier of the sensor.
sensorType	String	1	Type of the sensor. See note.
sensorPropertyList	Array(String)	1..N	It indicates the list of properties that the sensor can sense (see saref:Property in ETSI TS 103 264 [4], clause 4.3.8).
sensorCharacteristicList	Array(SensorCharacteristic)	0..N	The sensor' characteristics.
sensorPosition	Point	1	Geographical position of the sensor.
NOTE: When possible, it will leverage the sensor types defined by ETSI TS 103 264 [4], which contains the specification of the SAREF ontology and its extensions, clause 4.5. In case of types of sensors not included in the ones defined by SAREF, custom sensor types will be used.			

6.2.2 Type: SensorStatusInfo

This type represents the information related to the status of a sensor.

The attributes of the SensorStatusInfo shall follow the notations provided in Table 6.2.2-1.

Table 6.2.2-1: Definition of type SensorStatusInfo

Attribute name	Data type	Cardinality	Description
sensorIdentifier	String	1	Unique identifier of the sensor.
sensorStatusType	SensorStatusType	1	The status of the sensor, e.g. "ON", "OFF", "ERROR", "ONLINE", "OFFLINE (see saref:State in ETSI TS 103 264 [4], clause 4.3.5).
errorInformation	String	0..1	String explaining the error of the sensor.

6.2.3 Type: SensorData

This type represents the data that are gathered from a sensor with the related information.

The attributes of the SensorData shall follow the notations provided in Table 6.2.3-1.

Table 6.2.3-1: Definition of type SensorData

Attribute name	Data type	Cardinality	Description
sensorIdentifier	String	1	Unique identifier of the sensor.
data	String	1	Data measurement provided by the sensor.
dataFormat	String	1	Format of the data. See note.
dataUnitOfMeasure	String	1	Unit of Measure of the data (see saref:UnitOfMeasure in ETSI TS 103 264 [4], clause 4.3.8).
dataTimestamp	TimeStamp	0..1	Time instant at which the data have been collected.
NOTE: When possible, the data types defined by W3C [i.3] will be used, in case of data types not included in the ones defined by W3C, custom data types will be used.			

6.2.4 Type: SensorCharacteristicInfo

This type represents the characteristics of a sensor.

The attributes of the SensorCharacteristicInfo shall follow the notations provided in Table 6.2.4-1.

Table 6.2.4-1: Definition of type SensorCharacteristicInfo

Attribute name	Data type	Cardinality	Description
sensorIdentifier	String	1	Unique identifier of the sensor.
sensorCharacteristicList	Array(SensorCharacteristic)	0..N	The sensor' characteristics.

6.3 Subscription data types

6.3.1 Type: SensorDiscoveryEventSubscription

This type represents a subscription to notifications from the Sensor-sharing Service regarding events related to changes in sensor information.

The attributes of the SensorDiscoveryEventSubscription shall follow the indications provided in Table 6.3.1-1.

Table 6.3.1-1: Definition of type SensorDiscoveryEventSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "SensorDiscoveryEventSubscription".
callbackReference	Uri	0..1	URI exposed by the client on which to receive notifications via HTTP. See note 2.
requestTestNotification	Boolean	0..1	Set to TRUE by the MEC application to request a test notification via HTTP on the callbackReference URI, as specified in ETSI GS MEC 009 [3], clause 6.12a.
websockNotifConfig	WebsockNotifConfig	0..1	Provides details to negotiate and signal the use of a Websocket connection between the Sensor-sharing service and the MEC application for notifications. See note 2.
_links	Structure (inlined)	0..1	Hyperlink related to the resource. This shall be only included in the HTTP responses and in HTTP PUT requests.
>self	LinkType	1	Self-referring URI. The URI shall be unique within the SensorDiscoveryEventSubscription as it acts as an ID for the subscription.
sensorInfo	Structure (inlined)	0..N	Object containing the characteristics of the sensor(s) to be selected for the subscription.
>type	String	1	Type of the sensor. See note 3.
>sensorPropertyList	Array(String)	1..N	It indicates the list of properties that the sensor can sense (see saref:Property in ETSI TS 103 264 [4], clause 4.3.8).
>sensorCharacteristicList	Array(SensorCharacteristic)	0..N	The sensor' characteristics to be matched.
geographicalArea	AreaInfo	0..N	The parameters describing the area to subscribe.
expiryDeadline	TimeStamp	0..1	The expiration time of the subscription.

NOTE 1: Void.
NOTE 2: At least one of callbackReference and websockNotifConfig shall be provided by the MEC application. If both are provided, it is up to Sensor-sharing Service to select an alternative and return only that alternative in the response, as specified in ETSI GS MEC 009 [3], clause 6.12a.
NOTE 3: When possible, it will leverage the sensor types defined by ETSI TS 103 264 [4], which contains the specification of the SAREF ontology and its extensions [4], clause 4.5. In case of types of sensors not included in the ones defined by SAREF, custom sensor types will be used.

6.3.2 Type: SensorStatusSubscription

This type represents a subscription to notifications from the Sensor-sharing Service regarding changes to the sensor status.

The attributes of the SensorStatusSubscription shall follow the indications provided in Table 6.3.2-1.

Table 6.3.2-1: Definition of type SensorStatusSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "SensorStatusSubscription".
callbackReference	Uri	0..1	URI exposed by the client on which to receive notifications via HTTP. See note 2.
requestTestNotification	Boolean	0..1	Set to TRUE by the MEC application to request a test notification via HTTP on the callbackReference URI, as specified in ETSI GS MEC 009 [3], clause 6.12a.
websockNotifConfig	WebsockNotifConfig	0..1	Provides details to negotiate and signal the use of a Websocket connection between the Sensor-sharing service and the MEC application for notifications. See note 2.
_links	Structure (inlined)	0..1	Hyperlink related to the resource. This shall be only included in the HTTP responses and in HTTP PUT requests.
>self	LinkType	1	Self-referring URI. The URI shall be unique within the SensorStatusSubscription as it acts as an ID for the subscription.
sensorIdentifierList	Array(String)	1..N	Unique identifiers of the sensors.
expiryDeadline	TimeStamp	0..1	The expiration time of the subscription.
NOTE 1: Void. NOTE 2: At least one of callbackReference and websockNotifConfig shall be provided by the MEC application. If both are provided, it is up to Sensor-sharing Service to select an alternative and return only that alternative in the response, as specified in ETSI GS MEC 009 [3], clause 6.12a.			

6.3.3 Type: SensorDataSubscription

This type represents a subscription to notifications from the Sensor-sharing Service providing new data that are collected by the sensor(s).

The attributes of the SensorDataSubscription shall follow the indications provided in Table 6.3.3-1.

Table 6.3.3-1: Definition of type SensorDataSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "SensorDataSubscription".
callbackReference	Uri	0..1	URI exposed by the client on which to receive notifications via HTTP. See note 2.
requestTestNotification	Boolean	0..1	Set to TRUE by the MEC application to request a test notification via HTTP on the callbackReference URI, as specified in ETSI GS MEC 009 [3], clause 6.12a.
websockNotifConfig	WebsockNotifConfig	0..1	Provides details to negotiate and signal the use of a Websocket connection between the Sensor-sharing service and the MEC application for notifications. See note 2.
_links	Structure (inlined)	0..1	Hyperlink related to the resource. This shall be only included in the HTTP responses and in HTTP PUT requests.
>self	LinkType	1	Self-referring URI. The URI shall be unique within the SensorDataSubscription as it acts as an ID for the subscription.
sensorIdentifierList	Array(String)	1..N	Unique identifiers of the sensors.
expiryDeadline	TimeStamp	0..1	The expiration time of the subscription.
NOTE 1: Void. NOTE 2: At least one of callbackReference and websockNotifConfig shall be provided by the MEC application. If both are provided, it is up to Sensor-sharing Service to select an alternative and return only that alternative in the response, as specified in ETSI GS MEC 009 [3], clause 6.12a.			

6.3.4 Type: SubscriptionLinkList

This type represents a list of links related to currently existing subscriptions for the MEC application. This information is returned when sending a request to receive current subscriptions.

The attributes of the SubscriptionLinkList shall follow the indications provided in Table 6.3.4-1.

Table 6.3.4-1: Attributes of the SubscriptionLinkList

Attribute name	Data type	Cardinality	Description
_links	Structure (inlined)	1	List of hyperlinks related to the resource.
>self	LinkType	1	URI of this resource.
>subscriptions	Structure (inlined)	0..N	The MEC application's subscriptions.
>>href	Uri	1	The URI referring to the subscription.
>>subscriptionType	String	1	Type of the subscription. The string shall be set according to the "subscriptionType" attribute of the associated subscription data type defined in clauses 6.3.1, 6.3.2, 6.3.3: "SensorDiscoveryEventSubscription", "SensorStatusSubscription", "SensorDataSubscription".

6.4 Notifications data types

6.4.1 Type: SensorDiscoveryEventNotification

This type represents a notification from the Sensor-sharing Service regarding events related to changes in sensor information. The notification is sent by the Sensor-sharing Service when information related to a sensor is modified.

The attributes of the SensorDiscoveryEventNotification shall follow the indications provided in Table 6.4.1-1.

Table 6.4.1-1: Attributes of the SensorDiscoveryEventNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "SensorDiscoveryEventNotification".
timeStamp	TimeStamp	0..1	Time stamp.
sensorDiscoveryEventType	SensorDiscoveryEventType	1	The specific event triggering this notification, e.g. "NEW_SENSOR", "SENSOR_REMOVED". See note.
sensorDiscoveryInfo	SensorDiscoveryInfo	1..N	The information about the sensor(s) that are related to this discovery event notification.
_links	Structure(inlined)	1	Object containing hyperlinks related to the resource.
>subscription	LinkType	1	A link to the related subscription.
NOTE: The subscribed MEC application, after the subscription, can retrieve a complete list of the available sensors using the sensor discovery lookup procedure that is introduced in clause 5.3.2.			

6.4.2 Type: SensorStatusNotification

This type represents a notification from the Sensor-sharing Service to inform about the modifications in the sensors status. The notification is sent by the Sensor-sharing Service when the status of a sensor changes.

The attributes of the SensorStatusNotification shall follow the indications provided in Table 6.4.2-1.

Table 6.4.2-1: Attributes of the SensorStatusNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "SensorStatusNotification".
timeStamp	TimeStamp	0..1	Time stamp.
sensorStatusInfo	SensorStatusInfo	1..N	The information about the status of a sensor.
_links	Structure(inlined)	1	Object containing hyperlinks related to the resource.
>subscription	LinkType	1	A link to the related subscription.

6.4.3 Type: SensorDataNotification

This type represents a notification from the Sensor-sharing Service to provide new data that are collected by the sensors. The notification is sent by the Sensor-sharing Service when new data are available.

The attributes of the SensorDataNotification shall follow the indications provided in Table 6.4.3-1.

Table 6.4.3-1: Attributes of the SensorDataNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "SensorDataNotification".
timeStamp	TimeStamp	0..1	Time stamp.
sensorData	SensorData	1..N	The data gathered from the sensor(s) and related information.
_links	Structure(inlined)	1	Object containing hyperlinks related to the resource.
>subscription	LinkType	1	A link to the related subscription.

6.4.4 Type: ExpiryNotification

This type represents a notification from the Sensor-sharing Service with regards to expiry of the existing subscription. The notification is sent by the Sensor-sharing Service to send information about expiry of a subscription.

The attributes of the ExpiryNotification shall follow the indications provided in Table 6.4.4-1.

Table 6.4.4-1: Attributes of the ExpiryNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "ExpiryNotification".
timeStamp	TimeStamp	0..1	Time stamp of the notification.
_links	Structure (inlined)	1	List of hyperlinks related to the resource.
>subscription	LinkType	1	URI identifying the subscription which has expired.
expiryDeadline	TimeStamp	1	Time stamp of the notification expiry.

6.5 Referenced structured data types

6.5.1 Type: AreaInfo

This type represents the parameters that describe an area.

Table 6.5.1-1: Attributes of type AreaInfo

Attribute name	Data type	Cardinality	Description
shape	Enum(inlined)	1	The shape of the area monitored: 1 = CIRCLE. 2 = POLYGON.
points	Array(Point)	1..N	Shall include one point if the shape is CIRCLE. Shall include 3-15 points if the shape is POLYGON.
radius	UnsignedInt	0..1	Shall be present if the shape is CIRCLE. It is expressed in meters.

6.5.2 Type: Point

This type represents the geographical location of a point.

Table 6.5.2-1: Attributes of type Point

Attribute name	Data type	Cardinality	Description
latitude	Float	1	Location latitude, expressed in the range -90° to +90°.
longitude	Float	1	Location longitude, expressed in the range -180° to +180°.

6.5.3 Type: TimeStamp

This type represents a time stamp.

Table 6.5.3-1: Attributes of type TimeStamp

Attribute name	Data type	Cardinality	Description
seconds	Uint32	1	The seconds part of the time. Time is defined as Unix-time since January 1, 1970, 00:00:00 UTC.
nanoSeconds	Uint32	1	The nanoseconds part of the time. Time is defined as Unix-time since January 1, 1970, 00:00:00 UTC.

6.5.4 Type: SensorCharacteristic

This type represents a sensor' characteristic which details the type of sensor and its working functionalities, such as the model of the sensor, the sampling frequency, the operating range, the accuracy. Other sensor' characteristics are strictly inherent to the type of the sensors.

Table 6.5.4-1: Attributes of type SensorCharacteristic

Attribute name	Data type	Cardinality	Description
characteristicName	String	1	The name of the characteristic.
characteristicValue	String	1	The value of the characteristic.
characteristicUnitOfMeasure	String	0..1	The unit of measure of the characteristic (see saref:UnitOfMeasure in ETSI TS 103 264 [4], clause 4.3.8).

6.6 Referenced simple data types and enumerations

6.6.1 Type: LinkType

This type represents a type of link and may be referenced from data structures.

Table 6.6.1-1: Definition of type LinkType

Attribute name	Data type	Cardinality	Description
href	Uri	1	URI referring to a resource.

6.6.2 Enumeration: SensorDiscoveryEventType

This type represents specified event types for sensor discovery report.

Table 6.6.2-1: Enumeration SensorDiscoveryEventType

Type name	Description
"NEW_SENSOR"	A sensor, that is new to the MEC application, is available.
"SENSOR_REMOVED"	A sensor is not anymore available.
"SENSOR_UPDATE"	One or more properties of a sensor have been updated.

6.6.3 Enumeration: SensorStatusType

This type represents the status of the sensor. See saref:State in ETSI TS 103 264 [4], clause 4.3.5.

Table 6.6.3-1: Enumeration SensorStatusType

Type name	Description
"ON"	The sensor in on.
"OFF"	The sensor is off.
"STANDBY"	The sensor is in idle mode.
"ERROR"	The sensor is in error mode.
"ONLINE"	The sensor is online.
"OFFLINE"	The sensor is offline.

7 API definition

7.1 Introduction

This clause defines the resources and operations of the Sensor-sharing Service API.

7.2 Global definitions and resource structure

All resource URIs of this API shall have the following root:

{apiRoot}/{apiName}/{apiVersion}/

"apiRoot" and "apiName" are discovered using the service registry. It includes the scheme ("https"), host and optional port, and an optional prefix string. The "apiName" shall be set to "sens" and "apiVersion" shall be set to "v1" for the present document.

The API shall support HTTP over TLS (also known as HTTPS) using TLS version 1.2 (as defined by IETF RFC 5246 [6]). TLS 1.3 (including the new specific requirements for TLS 1.2 implementations) defined by IETF RFC 8446 [7] should be supported. HTTP without TLS shall not be used. Versions of TLS earlier than 1.2 shall neither be supported nor used.

TLS implementations should meet or exceed the security algorithm, key length and strength requirements specified in clause 6.2.3 of ETSI GS MEC 009 [3] (if TLS version 1.2 as defined by IETF RFC 5246 [6] is used) or clause 6.2.2 of ETSI GS MEC 009 [3] (if TLS version 1.3 as defined by IETF RFC 8446 [7] is used) of ETSI TS 133 210 [8] (Release 16 or later).

The content format of JSON shall be supported. The JSON format is signalled by the content type "application/json".

This API shall require the use of the OAuth 2.0 client credentials grant type according to IETF RFC 6749 [9] with bearer tokens according to IETF RFC 6750 [10]. See clause 6.16 of ETSI GS MEC 009 [3] for more information. How the token endpoint and client credentials are provisioned into the MEC applications is out of scope of the present document.

This API supports additional application-related error information to be provided in the HTTP response when an error occurs. See clause 6.15 of ETSI GS MEC 009 [3] for more information.

Figure 7.2-1 illustrates the resource URI structure of this API. Table 7.2-1 provides an overview of the resources defined by the present document, and the applicable HTTP methods.

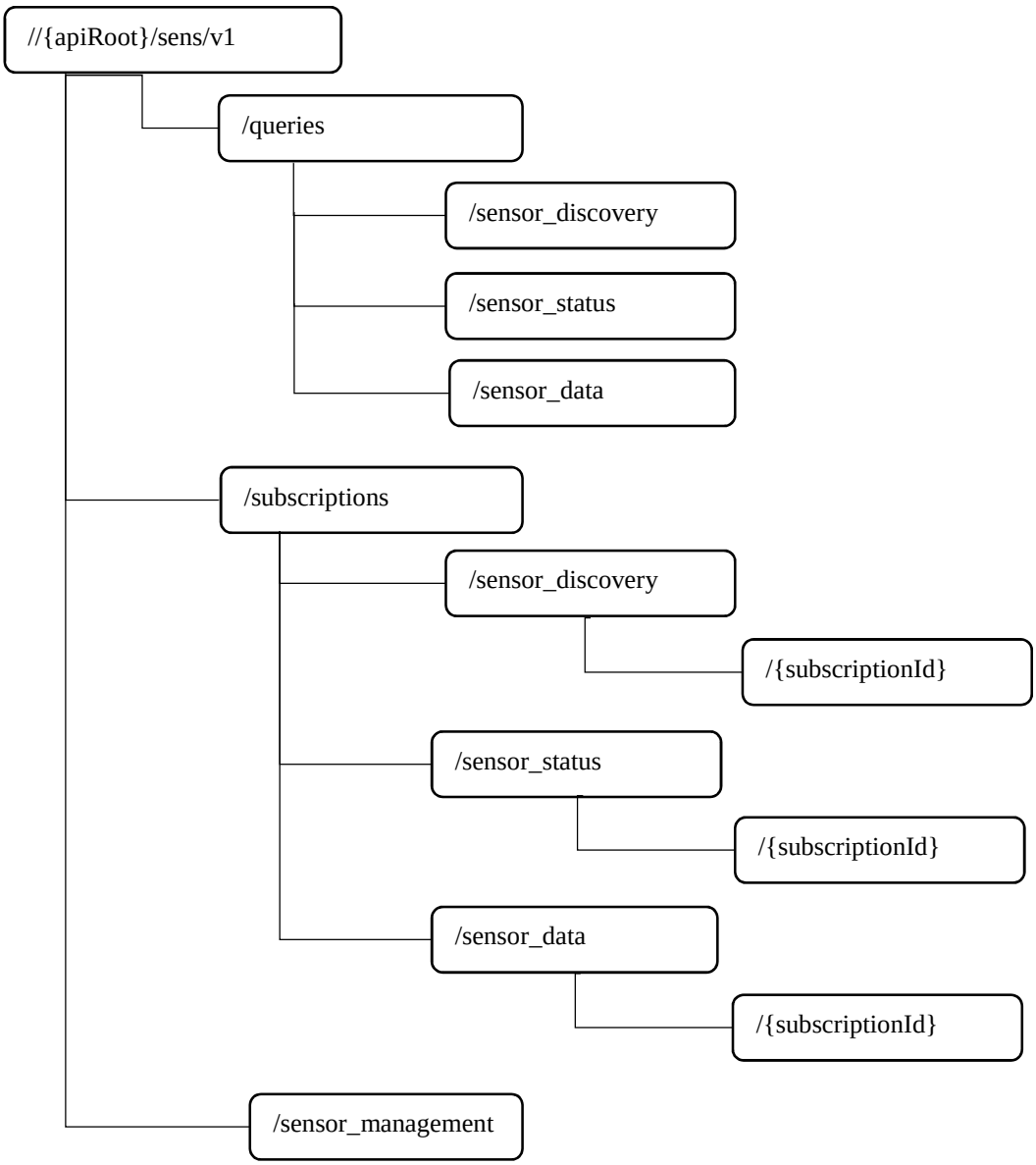

Figure 7.2-1: Resource URI structure of the Sensor-sharing Service API

Table 7.2-1: Resources and methods overview

Resource name	Resource URI	HTTP method	Meaning
Sensor discovery information	/queries/sensor_discovery	GET	Retrieve information about the available sensors.
Sensor status information	/queries/sensor_status	GET	Retrieve status information about specific sensors.
Sensor data	/queries/sensor_data	GET	Retrieve the last available data from specific sensors.
All sensor discovery subscriptions for a subscriber	/subscriptions/sensor_discovery	GET	Retrieve a list of active sensor discovery subscriptions for this subscriber.
		POST	Create a new sensor discovery subscription.
Existing subscription for sensor discovery	/subscriptions/sensor_discovery/{subscriptionId}	GET	Retrieve information on current specific sensor discovery subscription.
		PUT	Modify existing sensor discovery subscription by sending a new data structure.
		DELETE	Cancel the existing sensor discovery subscription.
All sensor status subscriptions for a subscriber	/subscriptions/sensor_status	GET	Retrieve a list of active sensor status subscriptions for this subscriber.
		POST	Create a new sensor status subscription.
Existing subscription for sensor discovery	/subscriptions/sensor_status/{subscriptionId}	GET	Retrieve information on current specific subscription.
		PUT	Modify existing subscription by sending a new data structure.
		DELETE	Cancel the existing subscription.
All sensor data subscriptions for a subscriber	/subscriptions/sensor_data	GET	Retrieve a list of active sensor discovery subscriptions for this subscriber.
		POST	Create a new sensor discovery subscription.
Existing subscription for sensor data	/subscriptions/sensor_data/{subscriptionId}	GET	Retrieve information on current specific subscription.
		PUT	Modify existing subscription by sending a new data structure.
		DELETE	Cancel the existing subscription.
Notification callback	Client provided callback reference	POST	Send a notification.
Sensor management	/sensor_management	GET	Retrieve the information about the characteristics that can be modified on specific sensors.
		PUT	Update the values of characteristics on specific sensors.

7.3 Resource: sensor_discovery

7.3.1 Description

This resource is queried to retrieve the information about the available sensors.

7.3.2 Resource definition

Resource URI: **{apiRoot}/sens/v1/queries/sensor_discovery**

This resource shall support the resource URI variables defined in Table 7.3.2-1.

Table 7.3.2-1: Resource URI variables for resource "sensor_discovery"

Name	Definition
apiRoot	See clause 7.2.

7.3.3 Resource methods

7.3.3.1 GET

The GET method is used to query the available sensors.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in Tables 7.3.3.1-1 and 7.3.3.1-2.

Table 7.3.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
sensorInfo	Structure (inlined)	0..N	Object containing the characteristics of the sensor(s) to be selected for the query.
>type	String	1	Type of the sensor. See note.
>sensorPropertyList	Array(String)	1..N	It indicates the list of properties that the sensor can sense (see saref:Property in ETSI TS 103 264 [4], clause 4.3.8).
>sensorCharacteristicList	Array(SensorCharacteristic)	0..N	The sensor' characteristics to be matched.
geographicalArea	AreaInfo	0..N	The parameters describing the area.
NOTE: When possible, it will leverage the sensor types defined by ETSI TS 103 264 [4], which contains the specification of the SAREF ontology and its extensions, clause 4.5. In case of types of sensors not included in the ones defined by SAREF, custom sensor types will be used.			

Table 7.3.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SensorDiscoveryInfo	0..N	200 OK	Upon success, a response body containing the available sensors is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.3.2 PUT

Not applicable.

7.3.3.3 PATCH

Not applicable.

7.3.3.4 POST

Not applicable.

7.3.3.5 DELETE

Not applicable.

7.4 Resource: sensor_discovery_subscriptions

7.4.1 Description

This resource contains various resources related to subscriptions for notifications related to the discovery of the available sensors.

7.4.2 Resource definition

Resource URI: **{apiRoot}/sens/v1/subscriptions/sensor_discovery**

This resource shall support the resource URI variables defined in Table 7.4.2-1.

Table 7.4.2-1: Resource URI variables for resource "sensor_discovery"

Name	Definition
apiRoot	See clause 7.2.

7.4.3 Resource methods

7.4.3.1 GET

The GET method is used to request information about the subscriptions related to sensor discovery for this requestor. Upon success, the response contains entity body with the list of links to the subscriptions related to sensor discovery that are present for the requestor.

This method shall support the URI query parameter, request and response data structures, and response codes, as specified in Tables 7.4.3.1-1 and 7.4.3.1-2.

Table 7.4.3.1-1: URI query parameter supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
sensorIdentifier	String	0..N	Unique identifiers of the sensors.

Table 7.4.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SubscriptionLinkList	1	200 OK	Upon success, a response body containing the list of links to requestor's sensor discovery subscriptions is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.3.2 PUT

Not applicable.

7.4.3.3 PATCH

Not applicable.

7.4.3.4 POST

The POST method is used to create a new subscription to sensor discovery notifications. Upon success, the response contains entity body describing the created subscription.

This method shall support the request and response data structures, and response codes, as specified in Table 7.4.3.4-1.

Table 7.4.3.4-1: Data structures supported by the POST request/response on this resource

	Data type	Cardinality	Remarks	
Request body	SensorDiscoveryEventSubscription	1	The entity body in the request contains data type of the specific sensor discovery subscription that is to be created, as defined in clause 6.3.1.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SensorDiscoveryEventSubscription	1	201 Created	Indicates successful resource creation, where the resource URI shall be returned in the HTTP Location header field. In the returned SensorDiscoveryEventSubscription structure, the created subscription is described using the appropriate data type as defined in clause 6.3.1.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	415 Unsupported Media Type	It is used to indicate that the server or the client does not support the content type of the entity body. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

Response body	ProblemDetails	0..1	422 Unprocessable Entity	It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if a JSON request body is syntactically correct but semantically incorrect, for example if the sensor ID is not in the correct format. This error condition can also occur if the capabilities required by the request are not supported. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.3.5 DELETE

Not applicable.

7.5 Resource: individual sensor_discovery_subscription

7.5.1 Description

This resource represents a subscription that the client has created to receive sensor discovery notifications.

7.5.2 Resource definition

Resource URI: **{apiRoot}/sens/v1/subscriptions/sensor_discovery/{subscription_id}**

This resource shall support the resource URI variables defined in Table 7.5.2-1.

Table 7.5.2-1: Resource URI variables for resource "individual sensor_discovery_subscription"

Name	Definition
apiRoot	See clause 7.2.
subscriptionId	Refers to created subscription, where the Sensor-sharing API allocates a unique resource name for this.

7.5.3 Resource methods

7.5.3.1 GET

The GET method is used to retrieve information about this subscription. Upon success, the response contains entity body with the data type describing the subscription.

This method shall support the request and response data structures, and response codes, as specified in Tables 7.5.3.1-1 and 7.5.3.1-2.

Table 7.5.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.5.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SensorDiscoveryEventSubscription	1	200 OK	Upon success, a response body containing data type describing the specific sensor discovery subscription is returned. The data type used is defined in clause 6.3.1.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.5.3.2 PUT

The PUT method is used to update the existing subscription. PUT method in this case has "replace" semantics. Upon successful operation, the target resource is updated with new Data Type received within the message body of the PUT request. This method shall support the URI query parameters, request and response data structures, and response codes, as specified in Tables 7.5.3.2-1 and 7.5.3.2-2.

Table 7.5.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.5.3.2-2: Data structures supported by the PUT request/response on this resource

	Data type	Cardinality	Response Codes	Remarks
Request body	SensorDiscoveryEventSubscription	1	New SensorDiscoveryEventSubscription is included as entity body of the request. The data type used is defined in clause 6.3.1.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SensorDiscoveryEventSubscription	1	200 OK	Upon success, a response body containing data type describing the updated sensor discovery subscription is returned. The data type used is defined in clause 6.3.1.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	422 Unprocessable Entity	It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if a JSON request body is syntactically correct but semantically incorrect, for example if the sensor ID is not in the correct format. This error condition can also occur if the capabilities required by the request are not supported. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

Response body	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.5.3.3 PATCH

Not applicable.

7.5.3.4 POST

Not applicable.

7.5.3.5 DELETE

The DELETE method is used to cancel the existing subscription. Cancellation can be made by deleting the resource that represents existing sensor discovery subscription. This method shall support the request and response data structures, and response codes, as specified in Tables 7.5.3.5-1 and 7.5.3.5-2.

Table 7.5.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.5.3.5-2: Data structures supported by the DELETE request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	n/a		204 No Content	Upon success, a response 204 No Content without any response body is returned.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.6 Resource: sensor_status

7.6.1 Description

This resource is queried to retrieve the status information of specific sensors.

7.6.2 Resource definition

Resource URI: **{apiRoot}/sens/v1/queries/sensor_status**

This resource shall support the resource URI variables defined in Table 7.6.2-1.

Table 7.6.2-1: Resource URI variables for resource "sensor_status"

Name	Definition
apiRoot	See clause 7.2.

7.6.3 Resource methods

7.6.3.1 GET

The GET method is used to query the status information of specific sensors.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in Tables 7.6.3.1-1 and 7.6.3.1-2.

Table 7.6.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
sensorIdentifier	String	1..N	Unique identifiers of the sensors to be queried. See note.
NOTE: In order to perform the sensor status lookup for a specific sensor or a group of specific sensors, the filtering parameter "sensorIdentifier" shall be used.			

Table 7.6.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SensorStatusInfo	1..N	200 OK	Upon success, a response body containing the status information of specific sensors is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.6.3.2 PUT

Not applicable.

7.6.3.3 PATCH

Not applicable.

7.6.3.4 POST

Not applicable.

7.6.3.5 DELETE

Not applicable.

7.7 Resource: sensor_status_subscriptions

7.7.1 Description

This resource contains various resources related to subscriptions for notifications related to sensor status.

7.7.2 Resource definition

Resource URI: **{apiRoot}/sens/v1/subscriptions/sensor_status**

This resource shall support the resource URI variables defined in Table 7.7.2-1.

Table 7.7.2-1: Resource URI variables for resource "sensor_status"

Name	Definition
apiRoot	See clause 7.2.

7.7.3 Resource methods

7.7.3.1 GET

The GET method is used to request information about the subscriptions related to sensor status for this requestor. Upon success, the response contains entity body with the list of links to the subscriptions related to sensor status that are present for the requestor.

This method shall support the URI query parameter, request and response data structures, and response codes, as specified in Tables 7.7.3.1-1 and 7.7.3.1-2.

Table 7.7.3.1-1: URI query parameter supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
sensorIdentifier	String	0..N	Unique identifiers of the sensors.

Table 7.7.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SubscriptionLinkList	1	200 OK	Upon success, a response body containing the list of links to requestor's sensor status subscriptions is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.7.3.2 PUT

Not applicable.

7.7.3.3 PATCH

Not applicable.

7.7.3.4 POST

The POST method is used to create a new subscription to sensor status notifications. Upon success, the response contains entity body describing the created subscription.

This method shall support the request and response data structures, and response codes, as specified in Table 7.7.3.4-1.

Table 7.7.3.4-1: Data structures supported by the POST request/response on this resource

	Data type	Cardinality	Remarks	
Request body	SensorStatusSubscription	1	The entity body in the request contains data type of the specific sensor status subscription that is to be created, as defined in clause 6.3.2.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SensorStatusSubscription	1	201 Created	Indicates successful resource creation, where the resource URI shall be returned in the HTTP Location header field. In the returned SensorStatusSubscription structure, the created subscription is described using the appropriate data type as defined in clause 6.3.2.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	415 Unsupported Media Type	It is used to indicate that the server or the client does not support the content type of the entity body. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

Response body	ProblemDetails	0..1	422 Unprocessable Entity	It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if a JSON request body is syntactically correct but semantically incorrect, for example if the sensor ID is not in the correct format. This error condition can also occur if the capabilities required by the request are not supported. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.7.3.5 DELETE

Not applicable.

7.8 Resource: individual sensor_status_subscription

7.8.1 Description

This resource represents a subscription that the client has created to receive sensor status notifications.

7.8.2 Resource definition

Resource URI: **{apiRoot}/sens/v1/subscriptions/sensor_status/{subscription_id}**

This resource shall support the resource URI variables defined in Table 7.8.2-1.

Table 7.8.2-1: Resource URI variables for resource "individual sensor_status_subscription"

Name	Definition
apiRoot	See clause 7.2.
subscriptionId	Refers to created subscription, where the Sensor-sharing API allocates a unique resource name for this.

7.8.3 Resource methods

7.8.3.1 GET

The GET method is used to retrieve information about this subscription. Upon success, the response contains entity body with the data type describing the subscription.

This method shall support the request and response data structures, and response codes, as specified in Tables 7.8.3.1-1 and 7.8.3.1-2.

Table 7.8.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.8.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	SensorStatusSubscription	1	200 OK	Upon success, a response body containing data type describing the specific sensor status subscription is returned. The data type used is defined in clause 6.3.2.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.8.3.2 PUT

The PUT method is used to update the existing subscription. PUT method in this case has "replace" semantics. Upon successful operation, the target resource is updated with new Data Type received within the message body of the PUT request. This method shall support the URI query parameters, request and response data structures, and response codes, as specified in Tables 7.8.3.2-1 and 7.8.3.2-2.

Table 7.8.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.8.3.2-2: Data structures supported by the PUT request/response on this resource

	Data type	Cardinality	Remarks	
Request body	SensorStatusSubscription	1	New SensorStatusSubscription is included as entity body of the request. The data type used is defined in clause 6.3.2.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SensorStatusSubscription	1	200 OK	Upon success, a response body containing data type describing the updated sensor status subscription is returned. The data type used is defined in clause 6.3.2.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	422 Unprocessable Entity	It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if a JSON request body is syntactically correct but semantically incorrect, for example if the sensor ID is not in the correct format. This error condition can also occur if the capabilities required by the request are not supported. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

Response body	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.8.3.3 PATCH

Not applicable.

7.8.3.4 POST

Not applicable.

7.8.3.5 DELETE

The DELETE method is used to cancel the existing subscription. Cancellation can be made by deleting the resource that represents existing sensor status subscription. This method shall support the request and response data structures, and response codes, as specified in Tables 7.8.3.5-1 and 7.8.3.5-2.

Table 7.8.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.8.3.5-2: Data structures supported by the DELETE request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	n/a		204 No Content	Upon success, a response 204 No Content without any response body is returned.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.9 Resource: sensor_data

7.9.1 Description

This resource is queried to retrieve the last sensor data from specific sensors.

7.9.2 Resource definition

Resource URI: **{apiRoot}/sens/v1/queries/sensor_data**

This resource shall support the resource URI variables defined in Table 7.9.2-1.

Table 7.9.2-1: Resource URI variables for resource "sensor_data"

Name	Definition
apiRoot	See clause 7.2

7.9.3 Resource methods

7.9.3.1 GET

The GET method is used to query the last sensor data from specific sensors.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in Tables 7.9.3.1-1 and 7.9.3.1-2.

Table 7.9.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
sensorIdentifier	String	1..N	Unique identifiers of the sensors to be queried. See note.
NOTE: In order to perform the sensor data lookup for a specific sensor or a group of specific sensors, the filtering parameter "sensorIdentifier" shall be used.			

Table 7.9.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SensorData	1..N	200 OK	Upon success, a response body containing the last sensor data measured by the sensors is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.9.3.2 PUT

Not applicable.

7.9.3.3 PATCH

Not applicable.

7.9.3.4 POST

Not applicable.

7.9.3.5 DELETE

Not applicable.

7.10 Resource: sensor_data_subscriptions

7.10.1 Description

This resource contains various resources related to subscriptions for notifications related to sensor data.

7.10.2 Resource definition

Resource URI: **{apiRoot}/sens/v1/subscriptions/sensor_data**

This resource shall support the resource URI variables defined in Table 7.10.2-1.

Table 7.10.2-1: Resource URI variables for resource "sensor_data"

Name	Definition
apiRoot	See clause 7.2.

7.10.3 Resource methods

7.10.3.1 GET

The GET method is used to request information about the subscriptions related to sensor data for this requestor. Upon success, the response contains entity body with the list of links to the subscriptions related to sensor data that are present for the requestor.

This method shall support the URI query parameter, request and response data structures, and response codes, as specified in Tables 7.10.3.1-1 and 7.10.3.1-2.

Table 7.10.3.1-1: URI query parameter supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
sensorIdentifier	String	0..N	Unique identifiers of the sensors.

Table 7.10.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SubscriptionLinkList	1	200 OK	Upon success, a response body containing the list of links to requestor's sensor data subscriptions is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.10.3.2 PUT

Not applicable.

7.10.3.3 PATCH

Not applicable.

7.10.3.4 POST

The POST method is used to create a new subscription to sensor data notifications. Upon success, the response contains entity body describing the created subscription.

This method shall support the request and response data structures, and response codes, as specified in Table 7.10.3.4-1.

Table 7.10.3.4-1: Data structures supported by the POST request/response on this resource

	Data type	Cardinality	Remarks	
Request body	SensorDataSubscription	1	The entity body in the request contains data type of the specific sensor data subscription that is to be created, as defined in clause 6.3.3.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SensorDataSubscription	1	201 Created	Indicates successful resource creation, where the resource URI shall be returned in the HTTP Location header field. In the returned SensorDataSubscription structure, the created subscription is described using the appropriate data type as defined in clause 6.3.3.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	415 Unsupported Media Type	It is used to indicate that the server or the client does not support the content type of the entity body. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

Response body	ProblemDetails	0..1	422 Unprocessable Entity	It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if a JSON request body is syntactically correct but semantically incorrect, for example if the sensor ID is not in the correct format. This error condition can also occur if the capabilities required by the request are not supported. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.10.3.5 DELETE

Not applicable.

7.11 Resource: individual sensor_data_subscription

7.11.1 Description

This resource represents a subscription that the client has created to receive sensor data notifications.

7.11.2 Resource definition

Resource URI: **{apiRoot}/sens/v1/subscriptions/sensor_data/{subscription_id}**

This resource shall support the resource URI variables defined in Table 7.11.2-1.

Table 7.11.2-1: Resource URI variables for resource "individual sensor_data_subscription"

Name	Definition
apiRoot	See clause 7.2.
subscriptionId	Refers to created subscription, where the Sensor-sharing API allocates a unique resource name for this.

7.11.3 Resource methods

7.11.3.1 GET

The GET method is used to retrieve information about this subscription. Upon success, the response contains entity body with the data type describing the subscription.

This method shall support the request and response data structures, and response codes, as specified in Tables 7.11.3.1-1 and 7.11.3.1-2.

Table 7.11.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.11.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	SensorDataSubscription	1	200 OK	Upon success, a response body containing data type describing the specific sensor data subscription is returned. The data type used is defined in clause 6.3.3.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.11.3.2 PUT

The PUT method is used to update the existing subscription. PUT method in this case has "replace" semantics. Upon successful operation, the target resource is updated with new Data Type received within the message body of the PUT request. This method shall support the URI query parameters, request and response data structures, and response codes, as specified in Tables 7.11.3.2-1 and 7.11.3.2-2.

Table 7.11.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.11.3.2-2: Data structures supported by the PUT request/response on this resource

	Data type	Cardinality	Remarks	
Request body	SensorDataSubscription	1	New SensorDataSubscription is included as entity body of the request. The data type used is defined in clause 6.3.3.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SensorDataSubscription	1	200 OK	Upon success, a response body containing data type describing the updated sensor data subscription is returned. The data type used is defined in clause 6.3.3.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	422 Unprocessable Entity	It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if a JSON request body is syntactically correct but semantically incorrect, for example if the sensor ID is not in the correct format. This error condition can also occur if the capabilities required by the request are not supported. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

Response body	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.11.3.3 PATCH

Not applicable.

7.11.3.4 POST

Not applicable.

7.11.3.5 DELETE

The DELETE method is used to cancel the existing subscription. Cancellation can be made by deleting the resource that represents existing sensor data subscription. This method shall support the request and response data structures, and response codes, as specified in Tables 7.11.3.5-1 and 7.11.3.5-2.

Table 7.11.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.11.3.5-2: Data structures supported by the DELETE request/response on this resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	n/a		204 No Content	Upon success, a response 204 No Content without any response body is returned.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.12 Resource: sensor_management

7.12.1 Description

This resource contains various resources used to manage the setting of the characteristics on a specific sensor or on a group of sensors.

7.12.2 Resource definition

Resource URI: **{apiRoot}/sens/v1/sensor_management**

This resource shall support the resource URI variables defined in Table 7.12.2-1.

Table 7.12.2-1: Resource URI variables for resource "sensor_management"

Name	Definition
apiRoot	See clause 7.2.

7.12.3 Resource methods

7.12.3.1 GET

The GET method is used to receive the sensor characteristics that can be modified on specific sensors.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in Tables 7.12.3.1-1 and 7.12.3.1-2.

Table 7.12.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
sensorIdentifier	String	1..N	Unique identifiers of the sensors.

Table 7.12.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SensorCharacteristicInfo	1..N	200 OK	Upon success, a response body containing the characteristics that can be set on the specific sensors are returned. This data type is described in clause: 6.2.4.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.

	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.12.3.2 PUT

The PUT method is used to update the value of the characteristics that can be set on a sensor or on a group of sensors. Upon successful operation, the values of sensors' characteristics are updated with the values received within the message body of the PUT request.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in Tables 7.12.3.2-1 and 7.12.3.2-2.

Table 7.12.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.12.3.2-2: Data structures supported by the PUT request/response on this resource

	Data type	Cardinality	Remarks	
Request body	SensorCharacteristicInfo	1..N	The list of sensor characteristics to be updated, with the new values to set, is included as entity body of the request. The data type used is defined in clause 6.2.4.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SensorCharacteristicInfo	1..N	200 OK	Upon success, a response body containing data type describing the updated sensor characteristics is returned. The data type used is defined in clause 6.2.4.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	422 Unprocessable Entity	It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if a JSON request body is syntactically correct but semantically incorrect, for example if the sensor ID is not in the correct format. This error condition can also occur if the capabilities required by the request are not supported. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

Response body	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.12.3.3 PATCH

Not applicable.

7.12.3.4 POST

Not applicable.

7.12.3.5 DELETE

Not applicable.

Annex A (informative): Complementary material for API utilization

To complement the definitions for each method and resource defined in the interface clauses of the present document, ETSI MEC ISG is providing for the user application lifecycle management API a supplementary description file compliant to the OpenAPI Specification [i.5].

In case of discrepancies between the supplementary description file and the related data structure definitions in the present document, the data structure definitions take precedence.

The supplementary description file, relating to the present document, is located at https://forge.etsi.org/rep/mec/gs046-ss-api.

History

Document history		
V3.1.1	April 2024	Publication
V3.2.1	August 2025	Publication