ETSI GS MEC 029 V2.2.1 (2022-01)

GROUP SPECIFICATION

Multi-access Edge Computing (MEC); Fixed Access Information 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-0029v221FixedAPI

Keywords

API, MEC, service

ETSI

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

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

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

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

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

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

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

Copyright Notification

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 Web server (https://ipr.etsi.org/).

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™** and **LTE™** 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 describes a MEC service on Fixed Access Information for Fibre (e.g. G-PON, XG-PON, NG-PON2, XGS-PON), Cable (DOCSIS 3.1), xDSL, and Point-to-Point Fibre Ethernet access networks. It describes the information flows, required information, and as applicable, specifies the necessary operations, data model and data format.

The present document also specifies the RESTful API.

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 at http://docbox.etsi.org/Reference.

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] Void.

[2] ETSI GS MEC 002: "Multi-access Edge Computing (MEC); Phase 2: Use Cases and Requirements".

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

NOTE: Available at https://tools.ietf.org/html/rfc6749.

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

NOTE: Available at https://tools.ietf.org/html/rfc6750.

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

NOTE: Available at https://tools.ietf.org/html/rfc5246.

[6] Void.

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

NOTE: Available at https://tools.ietf.org/html/rfc8446.

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

[i.1] ETSI GS MEC 009: "Multi-access Edge Computing (MEC); General principles, patterns and common aspects of MEC Service APIs".

[i.2] ETSI GS MEC 011: "Multi-access Edge Computing (MEC); Edge Platform Application Enablement".

[i.3] ETSI GS MEC 012: "Multi-access Edge Computing (MEC); Radio Network Information API".

[i.4] ETSI GS MEC 028: "Multi-access Edge Computing (MEC); WLAN Access Information API".

[i.5] Broadband Forum TR-106: "Data Model Template for TR-069-Enabled Devices".

[i.6] CableLabs® DOCSIS® 3.0: "Operations Support System Interface Specification", CM-SP-OSSIv3.0-C01-171207, December 7, 2017, Cable Television Laboratories, Inc.

[i.7] Recommendation ITU-T G.988 (11/2017): "ONU management and control interface (OMCI) specification".

[i.8] Recommendation ITU-T G.989.3 (10/2015): "40-Gigabit-capable passive optical networks (NG-PON2): Transmission convergence layer specification".

[i.9] Recommendation ITU-T G.984.3 (01/2014): "Gigabit-capable passive optical networks (G-PON): Transmission convergence layer specification".

[i.10] Recommendation ITU-T G.987.3 (01/2014): "10-Gigabit-capable passive optical networks (XG-PON): Transmission convergence (TC) layer specification".

[i.11] Recommendation ITU-T G.9807.1 (06/2016): "10-Gigabit-capable symmetric passive optical network (XGS-PON)".

[i.12] OpenAPI™ Specification.

NOTE: Available at https://github.com/OAI/OpenAPI-Specification.

[i.13] Protocol Buffers Language Specification.

NOTE 1: Available at https://developers.google.com/protocol-buffers/.

NOTE 2: Protocol Buffers Version 3 Language Specification is recommended as it is the official release at the time of publication.

[i.14] ETSI GS MEC 001: "Multi-access Edge Computing (MEC); Terminology".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GS MEC 001 [i.14] apply.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GS MEC 001 [i.14] and the following apply:

AC	Alternating Current
ADSL	Asymmetric Digital Subscriber Line
ANI	Access Node Interface
AQM	Active Queue Management
ASCII	American Standard Code for Information Interchange
BPI	Baseline Privacy Interface
CM	Cable Modem
CMTS	Cable Modem Termination System
DLS	Digital Subscriber Line
DOCSIS	Data Over Cable Service Interface Specification
EAE	Early Authentication and Encryption
FA	Fixed Access
FAI	Fixed Access Information
FAIS	Fixed Access Information Service
GFAST	G.fast (G stands for the ITU-T G series of recommendations)
GPON	Gigabit Passive Optical Network
NGPON	Next Generation Passive Optical Network
ONU	Optical Network Unit
PD	Powered Device
PoE	Power over Ethernet
PON	Passive Optical Network
SID	Service Identifier
UGS	Unsolicited Grant Service
VDSL	Very-high-bit-rate Digital Subscriber Line
xDSL	x Digital Subscriber Line (of any type)
XGPON	x Generation Passive Optical Network (also known as 10G-PON)
XG-PON	x Generation-Passive Optical Network
XGSPON	x Generation Symmetric Passive Optical Network

4 Overview

The present document specifies a Fixed Access Information (FAI) API to support the requirements defined for Multi-access Edge Computing in ETSI GS MEC 002 [2].

Clause 5 introduces how the Fixed Access Information Service (FAIS) supporting G-PON, XG-PON1, NG-PON2, XGS-PON and DOCSIS 3.1 may be used by the MEC applications and by the MEC platform. It describes information flows used for each.

The information that can be exchanged over the API is described in clause 6 which provides a detailed description of all the information elements that are used for each Fixed Access Network and how they are mapped into the REST operations.

Clause 7 describes the actual API, providing detailed information how information elements from each Fixed Access Network are mapped into the RESTful API design.

5 Description of the features (informative)

5.1 FAI service introduction

Multi-access Edge Computing allows running the MEC applications at the edge of the network where the environment is characterized by low latency, proximity to the end users, high bandwidth and exposure to location and up-to-date information from the underlying access networks. The information on current conditions from the fixed access is shared via FAIS.

FAIS is a service that provides the fixed access related information to service consumers within a MEC System. The FAIS is available for the authorized MEC applications and is discovered over the Mp1 reference point [i.2].

The FAI may be used by the MEC applications and the MEC platform to optimize the existing services and to provide new type of services that are based on up to date information from the fixed access possibly combined with the information such as Radio Network Information [i.3] or WLAN Information [i.4] from the other access technologies.

The following clauses describe how the service consumers interact with the FAIS over the FAI API to obtain contextual information from the fixed access network. The relevant sequence diagrams are presented.

5.2 Sequence diagrams

5.2.1 Introduction

The service consumers communicate with FAIS over the FAI API to get contextual information from the fixed access network. Both the MEC applications and the MEC platform may consume the FAIS; and both the MEC platform and the MEC applications may be the providers of the FAI.

The FAI API supports both queries and subscriptions (pub/sub mechanism) that are used over the RESTful API or over alternative transports such as message bus. Alternative transports are not specified in detail in the present document. For RESTful architectural style, the present document defines the HTTP protocol bindings.

5.2.2 Sending a request for the available FAI

Figure 5.2.2-1 shows a scenario where the service consumer (e.g. a MEC application or a MEC platform) sends a request to receive the available FAI that are relevant to the requested MEC application instance or the MEC platform. The response contains information of the available fixed access (e.g. Fibre (PON, XG-PON, NG-PON), Cable (DOCSIS 3.1), xDSL and Point-to-Point Fibre Ethernet access).

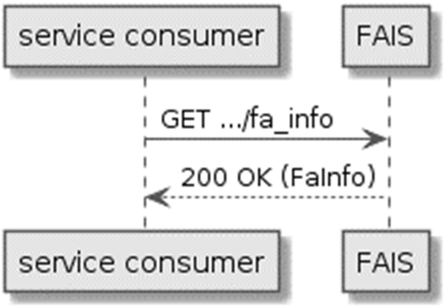

Figure 5.2.2-1: Flow of service consumer requesting the available FAI

A service consumer requesting the available FAI, as illustrated in figure 5.2.2-1, consists of the following steps:

1) Service consumer sends a GET request to the resource representing the available FAI.

2) FAIS responds with "200 OK" with the message body containing the FaInfo.

5.2.3 Sending a request for the device information

Figure 5.2.3-1 shows a scenario where the service consumer (e.g. a MEC application or a MEC platform) sends a request to receive the information of one or more devices connected to a fixed access network. The response contains information of the device(s).

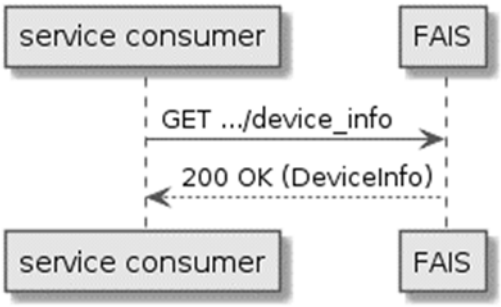

Figure 5.2.3-1: Flow of service consumer requesting the device information

A service consumer requesting the device information, as illustrated in figure 5.2.3-1, consists of the following steps:

1) Service consumer sends a GET request to the resource representing the device information.

2) FAIS responds with "200 OK" with the message body containing the DeviceInfo.

5.2.4 Sending a request for cable line information

Figure 5.2.4-1 shows a scenario where the service consumer (e.g. a MEC application or a MEC platform) sends a request to receive the information of the available cable line of a fixed access network. The response contains information of the line(s).

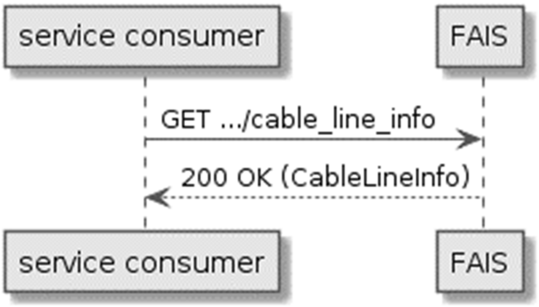

Figure 5.2.4-1: Flow of service consumer requesting the cable line information

A service consumer requesting the line information, as illustrated in figure 5.2.4-1, consists of the following steps:

1) Service consumer sends a GET request to the resource representing the line information.

2) FAIS responds with "200 OK" with the message body containing the LineInfo.

5.2.5 Sending a request for optical network information

Figure 5.2.5-1 shows a scenario where the service consumer (e.g. a MEC application or a MEC platform) sends a request to receive the information of the available information of an optical network (e.g. G-PON, XG-PON, NG-PON2, XGS-PON). The response contains information of the optical network.

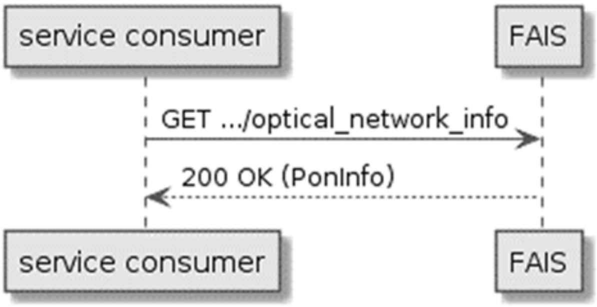

Figure 5.2.5-1: Flow of service consumer requesting the fibre line information

A service consumer requesting the line information, as illustrated in figure 5.2.5-1, consists of the following steps:

1) Service consumer sends a GET request to the resource representing the optical network information.

2) FAIS responds with "200 OK" with the message body containing the PonInfo.

5.2.6 REST based subscribe-notify model

5.2.6.1 Subscribing to event notifications

To receive notifications on selected FAI events, the service consumer creates a subscription to certain specific FAI event that is available at FAIS. Figure 5.2.6.1-1 shows a scenario where the service consumer uses REST based procedures to create a subscription for FAI event notifications.

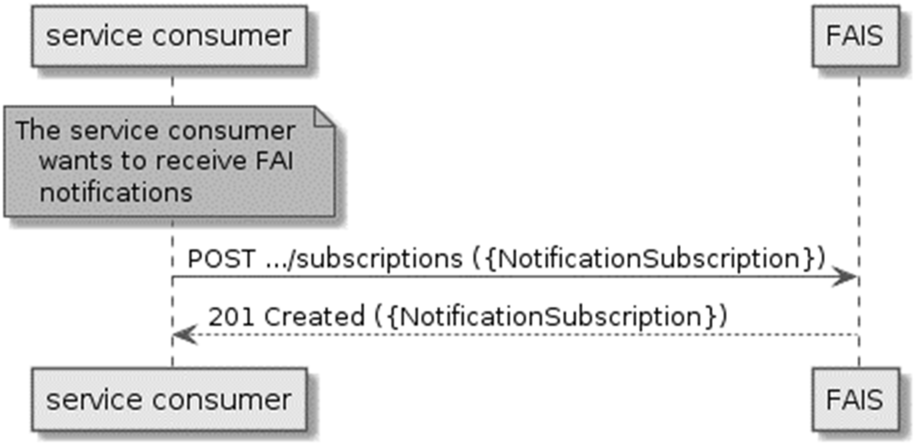
service consumer
FAIS
The service consumer wants to receive FAI notifications
POST .../subscriptions ({NotificationSubscription})
201 Created ({NotificationSubscription})
service consumer
FAIS

Figure 5.2.6.1-1: Flow of subscribing to the FAI event notifications

Subscribing to the FAI event notifications, as illustrated in figure 5.2.6.1-1, consists of the following steps.

When the service consumer wants to receive notifications about the FAI events, it creates a subscription to the FAI event notifications:

1) The service consumer sends a POST request with the message body containing the {NotificationSubscription} data structure to the resource representing FAI subscription. The variable {NotificationSubscription} is replaced with the data type specified for different FAI event subscriptions, and it defines the subscribed event, the filtering criteria and the address where the service consumer wishes to receive the FAI event notifications.

2) FAIS sends "201 Created" response with the message body containing the data structure specific to that FAI event subscription. The data structure contains the address of the resource created and the subscribed FAI event type. The address of the resource created is also contained in the message header.

5.2.6.2 Receiving notification on expiry of FAI event subscription

FAIS may define an expiry time for the FAI event subscription. In case expiry time is used, the time will be included in the {NotificationSubscription} data structure that is included in the response message to the subscription. Prior to the expiry, FAIS will also send a notification to the service consumer that owns the subscription.

Figure 5.2.6.2-1 shows a scenario where the service consumer receives a subscription expiry notification for the existing subscription.

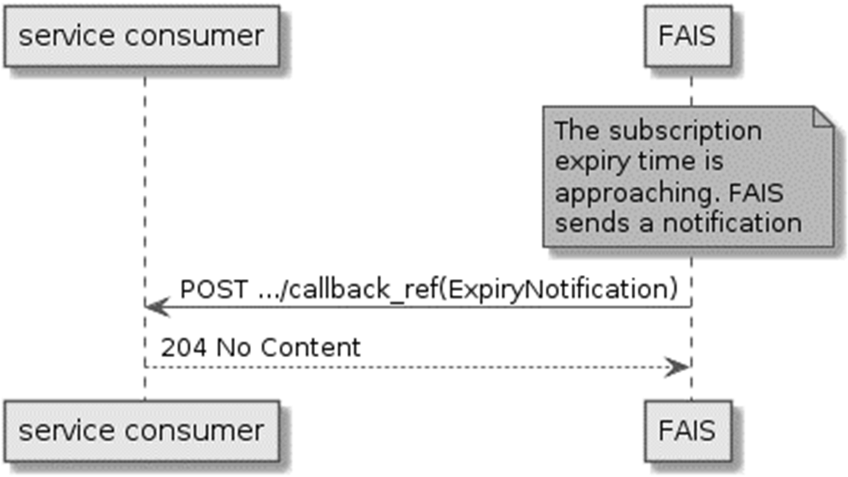
service consumer
FAIS
The subscription expiry time is approaching. FAIS sends a notification
POST .../callback_ref(ExpiryNotification)
204 No Content
service consumer
FAIS

Figure 5.2.6.2-1: Flow of FAIS sending a notification on expiry of the subscription

Sending a notification on expiry of the subscription, as illustrated in figure 5.2.6.2-1 consists of the following steps. If FAIS has defined an expiry time for the subscription, FAIS will send a notification prior to the expiry:

1) FAIS sends a POST request to the callback reference address included by the service consumer in the subscription request. The POST request contains a data structure ExpiryNotification.

2) Service consumer sends a "204 No Content" response.

5.2.6.3 Updating subscription for FAI event notifications

Figure 5.2.6.3-1 shows a scenario where the service consumer needs to update an existing subscription for a FAI event notification. The subscription update is triggered e.g. by the need to change the existing subscription, or due to the expiry of the subscription.

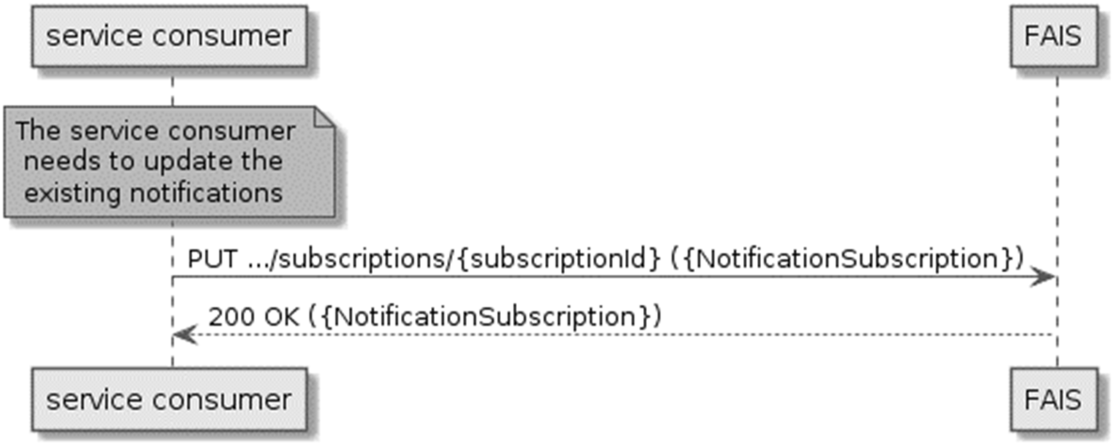
service consumer
FAIS
The service consumer needs to update the existing notifications
PUT .../subscriptions/{subscriptionId} ({NotificationSubscription})
200 OK ({NotificationSubscription})
service consumer
FAIS

Figure 5.2.6.3-1: Flow of service consumer updating subscription for FAI event notifications

Updating subscription for FAI event notifications, as illustrated in figure 5.2.6.3-1, consists of the following steps.

When the service consumer needs to modify an existing subscription for FAI event notifications, it can update the corresponding subscription as follows:

1) Service consumer updates the subscription resource by sending a PUT request to the resource representing the FAI event subscription that was created with the modified data structure specific to that FAI event subscription.

2) FAIS returns "200 OK" with the message body containing the accepted data structure specific to that FAI event subscription.

5.2.6.4 Unsubscribing from FAI event notifications

When the service consumer does not want to receive notifications anymore after subscribing to FAI events, the service consumer unsubscribes from the FAI event notifications. Figure 5.2.6.4-1 shows a scenario where the service consumer uses REST based procedures to delete the subscription for FAI event notifications.

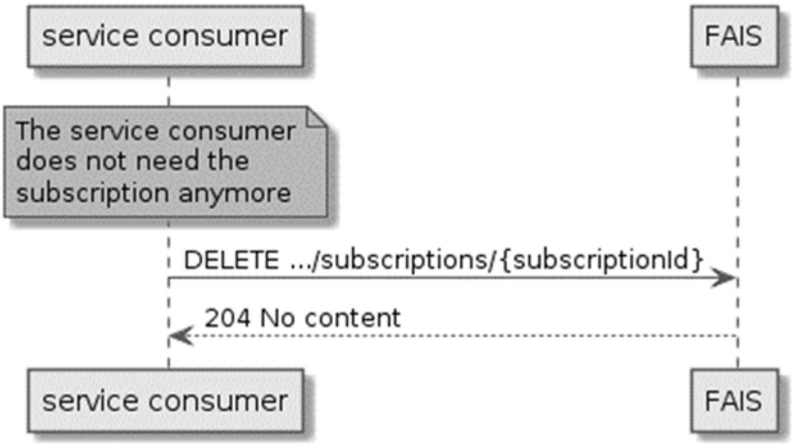
service consumer
FAIS
The service consumer does not need the subscription anymore
DELETE .../subscriptions/{subscriptionId}
204 No content
service consumer
FAIS

Figure 5.2.6.4-1: Flow of unsubscribing from the FAI event notifications

Unsubscribing from the FAI event notifications, as illustrated in figure 5.2.6.4-1, consists of the following steps:

When the service consumer does not want to receive the notifications anymore, it can unsubscribe from the FAI notification events by deleting the subscription:

1) Service consumer sends a DELETE request to the resource representing the FAI event subscription that was created.

2) FAIS sends "204 No content" response.

5.2.7 Receiving FAI event notifications about the ONU alarms

Figure 5.2.7-1 presents the scenario where the FAIS sends FAI event notification on the ONU alarms to the service consumer.

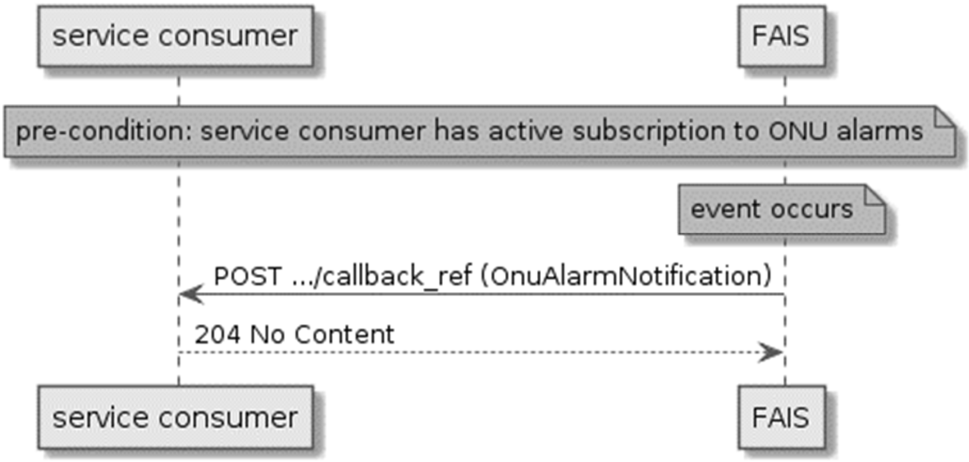
service consumer
FAIS
pre-condition: service consumer has active subscription to ONU alarms
event occurs
POST .../callback_ref (OnuAlarmNotification)
204 No Content
service consumer
FAIS

Figure 5.2.7-1: Flow of receiving FAI event notifications on the ONU alarms

Receiving FAI event notifications on ONU alarms, as illustrated in figure 5.2.7-1, consists of the following steps:

1) FAIS sends a POST request with the message body containing the OnuAlarmNotification data structure to the callback reference address included by the service consumer in the FAI ONU alarm event subscription.

2) Service consumer sends a "204 No Content" response to the FAIS.

5.2.8 Receiving FAI event notifications about the device information

Figure 5.2.8-1 presents the scenario where the FAIS sends FAI event notification on the abnormal operational status of the device to the service consumer.

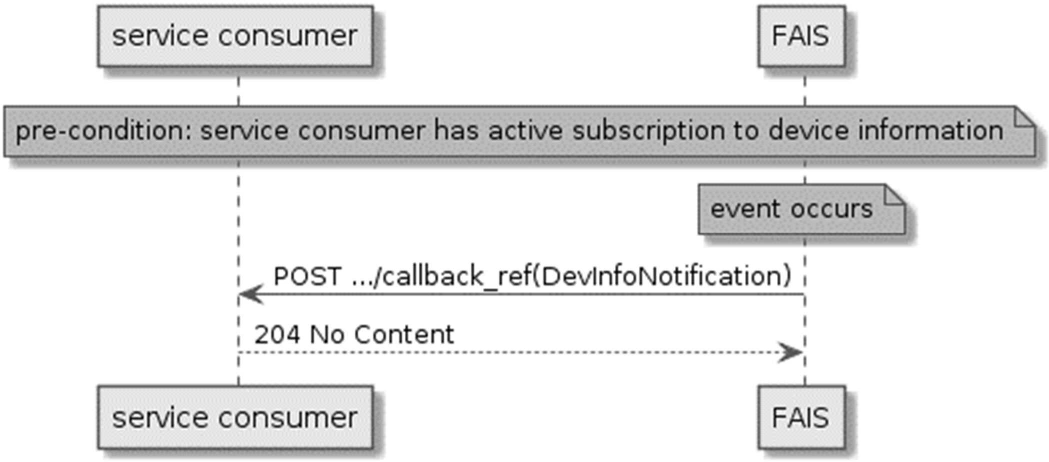
service consumer
FAIS
pre-condition: service consumer has active subscription to device information
event occurs
POST .../callback_ref(DevInfoNotification)
204 No Content
service consumer
FAIS

Figure 5.2.8-1: Flow of receiving FAI event notifications on the device information

Receiving FAI event notifications on device information, as illustrated in figure 5.2.8-1, consists of the following steps:

1) FAIS sends a POST request with the message body containing the DevInfoNotification data structure to the callback reference address included by the service consumer in the FAI device information event subscription.

2) Service consumer sends a "204 No Content" response to the FAIS.

5.2.9 Receiving FAI event notifications about the cable modem connectivity state

Figure 5.2.9-1 presents the scenario where the FAIS sends FAI event notification on the cable modem connectivity state to the service consumer.

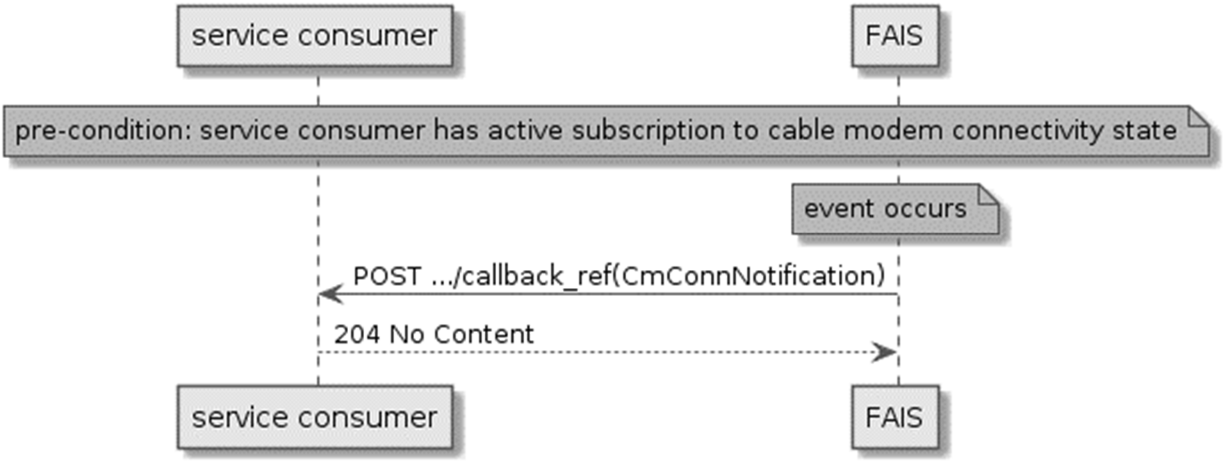
service consumer
FAIS
pre-condition: service consumer has active subscription to cable modem connectivity state
event occurs
POST .../callback_ref(CmConnNotification)
204 No Content
service consumer
FAIS

Figure 5.2.9-1: Flow of receiving FAI event notifications on the cable modem connectivity state

Receiving FAI event notifications on device information, as illustrated in figure 5.2.9-1, consists of the following steps:

1) FAIS sends a POST request with the message body containing the CmConnNotification data structure to the callback reference address included by the service consumer in the FAI cable modem connectivity state event subscription.

2) Service consumer sends a "204 No Content" response to the FAIS.

5.2.10 Receiving FAI event notifications about the ANI alarms

Figure 5.2.10-1 presents the scenario where the FAIS sends FAI event notification on the ANI alarms to the service consumer.

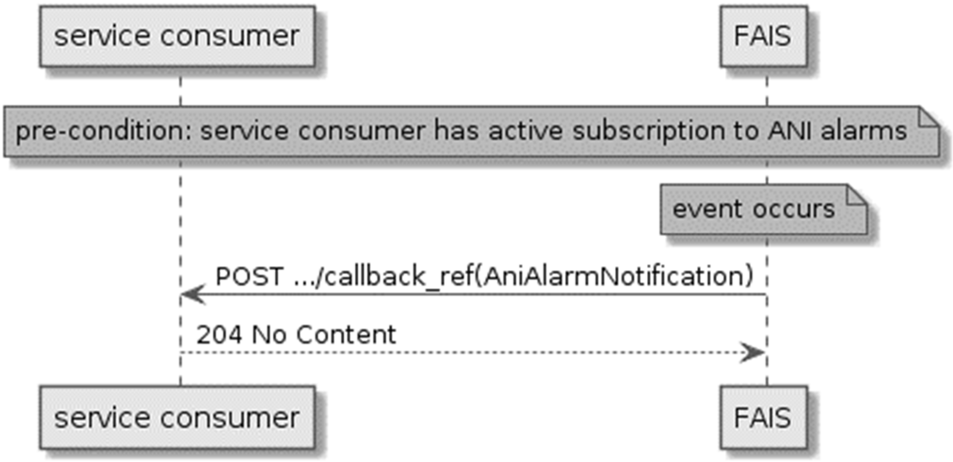
service consumer
FAIS
pre-condition: service consumer has active subscription to ANI alarms
event occurs
POST .../callback_ref(AniAlarmNotification)
204 No Content
service consumer
FAIS

Figure 5.2.10-1: Flow of receiving FAI event notifications on the ANI alarms

Receiving FAI event notifications on ANI alarms, as illustrated in figure 5.2.10-1, consists of the following steps:

1) FAIS sends a POST request with the message body containing the AniAlarmNotification data structure to the callback reference address included by the service consumer in the FAI ANI alarm event subscription.

2) Service consumer sends a "204 No Content" response to the FAIS.

6 Data model

6.1 Introduction

The following clauses provide the description of the data model.

6.2 Resource data types

6.2.1 Introduction

This clause defines data structures that shall be used in the resource representations.

6.2.2 Type: FaInfo

This type represents the information of a fixed access network.

The attributes of the FaInfo shall follow the notations provided in table 6.2.2-1.

Table 6.2.2-1: Attributes of the FaInfo

Attribute name	Data type	Cardinality	Description
timeStamp	TimeStamp	0..1	Time stamp.
customerPremisesInfo	CpInfo	1	The physical location of a customer site.
lastMileTech	Enum	1	An informative field identifying the last mile access technology used. The valid values are: 1 = ADSL. 2 = VDSL. 3 = GPON. 4 = XGPON. 5 = NGPON2. 6 = XGSPON. 7 = GFAST. 8 = P2PEthernet.
interfaceType	Enum	1	The physical interface used for the end customer site: 1 = 100BASE-TX. 2 = 1000BASE-TX. 3 = 1000BASE-LX. 4 = 1000BASELX10. 5 = 1000BASEBX10. 6 = 1000BASE-LH. 7 = 1000Base-ZX. 8 = ADSL-RJ11. 9 = VDSL-RJ11. 10 = GPON.
dsbw	Integer	0..1	The bandwidth (in Mbps) from the network towards the customer site.
usbw	Integer	0..1	The bandwidth (in Mbps) from the customer site towards the network.
latency	Integer	0..1	Maximum baseline latency (in ms) between customer site and service edge node.

6.2.3 Type: DeviceInfo

This type represents the information of the device that is connected to a fixed access network. The granularity of the device information may be adjusted based on parameters such as information per device, per gateway or per multiple gateways.

The attributes of the DeviceInfo shall follow the notations provided in table 6.2.3-1, as defined in [i.5].

Table 6.2.3-1: Attributes of the DeviceInfo

Attribute name	Data type	Cardinality	Description
timeStamp	TimeStamp	0..1	Time stamp.
gwId	String	1	Information (typically the serial number) to identify an Internet Gateway Device through which the customer premises device is connected. This value shall remain fixed over the lifetime of the device, including across firmware updates.
deviceId	String	1	Typically, the serial number of the device. This value shall remain fixed over the lifetime of the device, including across firmware updates.
deviceStatus	Enum	1	Current operational status of the device. Enumeration of: 1 = Up. 2 = Initializing. 3 = Error. 4 = Disabled.
upTime	Integer	1	Time in seconds since the device was last restarted.
iPConnectionUpTime	Integer	1	The time in seconds that the IP interface has been connected.
totalBytesSent	Integer	1	Total number of IP payload bytes sent since the device was last restarted.
totalBytesReceived	Integer	1	Total number of IP payload bytes received since the device was last restarted.
totalPacketsSent	Integer	1	Total number of packets sent since the device was last restarted.
totalPacketsReceived	Integer	1	Total number of packets received since the device was last restarted.
iPPingDiagnostics	IPPingDiagnostics	0..1	The result of an IP-layer ping test.
traceRouteDiagnostics	TraceRouteDiagnostics	0..1	The result of an IP-layer trace-route test.
downloadDiagnostics	DownloadDiagnostics	0..1	The result of a HTTP and FTP DownloadDiagnostics Test.
uploadDiagnostics	UploadDiagnostics	0..1	The result of a HTTP and FTP UploadDiagnostics Test.

6.2.4 Type: CableLineInfo

This type represents the information of the cable line of a fixed access network.

The attributes of the CableLineInfo shall follow the notations provided in table 6.2.4-1, as defined in [i.6].

Table 6.2.4-1: Attributes of the CableLineInfo

Attribute name	Data type	Cardinality	Description
timeStamp	TimeStamp	0..1	Time stamp.
customerPremisesInfo	CpInfo	1	The physical location of a customer site.
cmId	String	1	Information (typically the serial number) to identify a Cable Modem at subscriber locations intended for use in conveying data communications on a cable data system. This value SHALL remain fixed over the lifetime of the device, including across firmware updates.
cmStatus	Structure (inlined)	0..1	It provides CM connectivity status information of the CM.
>ifIndex	String	1	It denotes the MAC Domain interface index of the CM.

Attribute name	Data type	Cardinality	Description
>cmRegState	Enum	1	It defines the CM connectivity state. Enumeration of: 1 = other. 2 = notReady. 3 = notSynchronized. 4 = phySynchronized. 5 = usParametersAcquired. 6 = rangingComplete. 7 = dhcpV4Complete. 8 = todEstablished. 9 = securityEstablished. 10 = configFileDownloadComplete. 11 = registrationComplete. 12 = operational. 13 = accessDenied. 14 = eaeInProgress. 15 = dhcpv4InProgress. 16 = dhcpv6InProgress. 17 = dhcpV6Complete. 18 = registrationInProgress. 19 = bpiInit. 20 = forwardingDisabled. 21 = dsTopologyResolutionInProgress. 22 = rangingInProgress. 23 = rfMuteAll.
>resets	Integer	1	It denotes the number of times the CM reset or initialized this interface.
>lostSyncs	Integer	1	It denotes the number of times the CM lost synchronization with the downstream channel.
>invalidRegRsps	Integer	1	It denotes the number of times the CM received invalid registration response messages.
>energyMgt1x1OperStatus	Boolean	1	It indicates whether the CM is currently operating in Energy Management 1x1 Mode.
>emDlsOperStatus	Boolean	1	It indicates whether the CM is currently operating in Energy Management DLS Mode.
cmDpvStats	Structure (inlined)	0..1	It represents the DOCSIS Path Verify Statistics collected in the cable modem device.
>ifIndex	String	1	It represents the interface Index of the Downstream Interface where the measurements are taken.
>lastMeasLatency	Integer	1	The last latency measurement.
>lastMeasTime	TimeStamp	1	The last measurement time of the last latency measurement.
>minLatency	Integer	1	The minimum latency measurement.
>maxLatency	Integer	1	The maximum latency measurement.
>avgLatency	Integer	1	The average latency measurement.
>numMeas	Integer	1	The number of latency measurements made.
serviceFlowStats	Structure (inlined)	0..1	It describes statistics associated with the Service Flows in a managed device.
>ifIndex	String	1	It represents the interface index of the MAC Domain of the Service Flow.
>serviceFlowInfo	Structure (inlined)	1..N	It represents the information of a Service Flow.
>>serviceFlowId	Integer	1	It represents an identifier assigned to a Service Flow by CMTS within a MAC Domain.
>>pkts	Integer	1	For outgoing Service Flows, this attribute counts the number of Packet Data PDUs forwarded to this Service Flow. For incoming upstream CMTS service flows, this attribute counts the number of Packet Data PDUs actually received on the Service Flow identified by the SID for which the packet was scheduled.

Attribute name	Data type	Cardinality	Description
>>timeCreated	TimeStamp	1	It indicates the time when the service flow was created.
>>timeActive	Integer	1	It indicates the number of seconds that the service flow has been active.
>>policedDropPkts	Integer	1	For upstream service flows, this attribute counts the number of Packet Data PDUs classified to this service flow dropped due to: 1) exceeding the selected Buffer Size for the service flow; or 2) UGS packets dropped due to exceeding the Unsolicited Grant Size with a Request/Transmission policy that requires such packets to be dropped.
>>policedDelayPkts	Integer	1	It counts only outgoing packets delayed in order to maintain the Maximum Sustained Traffic Rate.
>>aqmDroppedPkts	Integer	1	For upstream service flows on which AQM is enabled, this attribute counts the number of Packet Data PDUs classified to this service flow dropped due to Active Queue Management drop decisions.

6.2.5 Type: PonInfo

This type represents the information of the of an optical network (e.g. G-PON, XG-PON, NG-PON2, XGS-PON).

The attributes of the PonInfo shall follow the notations provided in table 6.2.5-1, as defined in [i.7].

Table 6.2.5-1: Attributes of the PonInfo

Attribute name	Data type	Cardinality	Description
timeStamp	TimeStamp	0..1	Time stamp.
customerPremisesInfo	CpInfo	1	The physical location of a customer site.
ponSYS_ID	String	1	The 20-bit identity of the optical system within a certain domain. This is a reference value set by the OSS.
onuId	String	1	Information to identify an Optical Network Unit.
ponTech	Enum	1	An informative field identifying the optical technology used. The valid values are: 1 = GPON. 2 = XGPON. 3 = NGPON2. 4 = XGSPON.
operationalState	Enum	1	It reports whether the ONU is currently capable of performing its function. Valid values are: 0 = enabled 1 = disabled
dsRate	Enum	1	Downstream line rate, valid values are: 1 = 2,48832 Gbit/s: G-PON [i.9]; or NG-PON2 option 2 [i.8]. 2 = 9,95328 Gbit/s: XG-PON [i.10]; or XGS-PON [i.11]; or NG-PON2 option 1 [i.8].
usRate	Enum	1	Upstream line rate, valid values are: 1 = 1,24416 Gbit/s: G-PON option 1 [i.9]. 2 = 2,48832 Gbit/s: G-PON option 2 [i.9]; or XG-PON [i.10]; or NG-PON2 option 2 [i.8]. 3 = 9,95328 Gbit/s: XGS-PON [i.11]; or NG-PON2 option 1 [i.8].

6.3 Subscription data types

6.3.1 Introduction

This clause defines data structures for subscriptions.

6.3.2 Type: OnuAlarmSubscription

This type represents a subscription to ONU alarm notifications from FAIS.

The attributes of the OnuAlarmSubscription shall follow the indications provided in table 6.3.2-1, as defined in Recommendation ITU-T G.988 [i.7].

Table 6.3.2-1: Attributes of the OnuAlarmSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "OnuAlarmSubscription".
callbackReference	Uri	0..1	URI exposed by the client on which to receive notifications via HTTP. See note.
requestTestNotification	Boolean	0..1	Shall be set to TRUE by the service consumer to request a test notification via HTTP on the callbackReference URI, as described in ETSI GS MEC 009 [i.1], clause 6.12a. Default: FALSE.
websockNotifConfig	WebsockNotifConfig	0..1	Provides details to negotiate and signal the use of a Websocket connection between FAIS and the service consumer for notifications. See note.
_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 FAI API as it acts as an ID for the subscription.
filterCriteriaOnuAlarm	Structure (inlined)	1	List of filtering criteria for the subscription. Any filtering criteria from below, which is included in the request, shall also be included in the response.
>customerPremisesInfo	CpInfo	0..N	0 to N physical locations of the customer sites.
>onuId	String	1..N	1 to N unique identifiers for the optical network units. Typical it is the unique serial number for each ONU, which contains the vendor ID and version number. The first four bytes are an ASCII-encoded four-letter vendor ID. The second four bytes are a binary encoded serial number, under the control of the ONU vendor.

Attribute name	Data type	Cardinality	Description
>alarms	Enum	0..N	In case alarms is not included in the subscription request, the default value -1 = All shall be used and included in the response: 0 = EquipmentAlarm. (Functional failure on an internal interface.) 1 = PoweringAlarm. (Loss of external power to battery backup unit. This alarm is typically derived through an external interface to a battery backup unit, and indicates that AC is no longer available to maintain battery charge.) 2 = BatteryMissing. (Battery is provisioned but missing.) 3 = BatteryFailure. (Battery is provisioned and present but cannot recharge.) 4 = BatteryLow. (Battery is provisioned and present but its voltage is too low.) 5 = PhysicalIntrusion. (Applies if the ONU supports detection such as door or box open.) 6 = OnuSelfTestFailure. (ONU has failed autonomous self-test.) 7 = DyingGasp. (ONU is powering off imminently due to loss of power to the ONU itself.) 8 = TemperatureYellow. (No service shutdown at present, but the circuit pack is operating beyond its recommended range.) 9 = TemperatureRed. (Some services have been shut down to avoid equipment damage.) 10 = VoltageYellow. (No service shutdown at present, but the line power voltage is below its recommended minimum.) 11 = VoltageRed. (Some services have been shut down to avoid power collapse.) 12 = OnuManualPowerOff. (The ONU is shutting down because the subscriber has turned off its power switch.) 13 = InvImage. (Software image is invalid.) 14 = PseOverloadYellow. (Indicates that the ONU is nearing its maximum ability to supply the known PoE demand of the attached PDs. The thresholds for declaring and clearing this alarm are vendor-specific.) 15 = PseOverloadRed. (Indicates that the ONU is unable to supply all of the PoE demand of the attached PDs and has removed or reduced power to at least one PD.) -1 = All.
expiryDeadline	TimeStamp	0..1	Time stamp.
NOTE: At least one of callbackReference and websockNotifConfig shall be provided by the service consumer. If both are provided, it is up to FAIS to choose an alternative and return only that alternative in the response, as described in ETSI GS MEC 009 [i.1], clause 6.12a.			

6.3.3 Type: DevInfoSubscription

This type represents a subscription to device information notifications from FAIS.

The attributes of the DevInfoSubscription shall follow the indications provided in table 6.3.3-1.

Table 6.3.3-1: Attributes of the DevInfoSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "DevInfoSubscription".
callbackReference	Uri	0..1	URI exposed by the client on which to receive notifications via HTTP. See note.
requestTestNotification	Boolean	0..1	Shall be set to TRUE by the service consumer to request a test notification via HTTP on the callbackReference URI, as described in ETSI GS MEC 009 [i.1], clause 6.12a. Default: FALSE.
websockNotifConfig	WebsockNotifConfig	0..1	Provides details to negotiate and signal the use of a Websocket connection between FAIS and the service consumer for notifications. See note.
_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 FAI API as it acts as an ID for the subscription.
filterCriteriaDevInfo	Structure (inlined)	1	List of filtering criteria for the subscription. Any filtering criteria from below, which is included in the request, shall also be included in the response.
>gwId	String	0..N	The identifier of an Internet Gateway Device through which the customer premises device is connected.
>deviceId	String	0..N	The device identifier.
>deviceErrStatus	Enum	0..N	The abnormal operational status of the device. Enumeration of: 1 = Error. 2 = Disabled.
expiryDeadline	TimeStamp	0..1	Time stamp.
NOTE: At least one of callbackReference and websockNotifConfig shall be provided by the service consumer. If both are provided, it is up to FAIS to choose an alternative and return only that alternative in the response, as described in ETSI GS MEC 009 [i.1], clause 6.12a.			

6.3.4 Type: CmConnSubscription

This type represents a subscription to cable modem connectivity state notifications from FAIS.

The attributes of the CmConnSubscription shall follow the indications provided in table 6.3.4-1.

Table 6.3.4-1: Attributes of the CmConnSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "CmConnSubscription".
callbackReference	Uri	0..1	URI exposed by the client on which to receive notifications via HTTP. See note 1.
requestTestNotification	Boolean	0..1	Shall be set to TRUE by the service consumer to request a test notification via HTTP on the callbackReference URI, as described in ETSI GS MEC 009 [i.1], clause 6.12a. Default: FALSE.
websockNotifConfig	WebsockNotifConfig	0..1	Provides details to negotiate and signal the use of a Websocket connection between FAIS and the service consumer for notifications. See note 1.
_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 FAI API as it acts as an ID for the subscription.

Attribute name	Data type	Cardinality	Description
filterCriteriaCmConn	Structure (inlined)	1	List of filtering criteria for the subscription. Any filtering criteria from below, which is included in the request, shall also be included in the response.
>customerPremisesInfo	CpInfo	0..1	The physical location of a customer site.
>cmIf	Structure (inlined)	0..N	
>>cmId	String	1	The identifier of a Cable Modem at subscriber locations intended for use in conveying data communications on a cable data system.
>>ifIndex	String	1..N	The MAC Domain interface index of the CM.
>>cmRegState	Enum	1..N	It indicates the CM connectivity state. See note 2 for detail. Enumeration of: 1 = notReady. 2 = notSynchronized. 3 = phySynchronized. 4 = dsTopologyResolutionInProgress. 5 = usParametersAcquired. 6 = rangingInProgress. 7 = rangingComplete. 8 = eaeInProgress. 9 = dhcpv4InProgress. 10 = dhcpv6InProgress. 11 = dhcpV4Complete. 12 = dhcpV6Complete. 13 = todEstablished. 14 = securityEstablished. 15 = configFileDownloadComplete. 16 = registrationInProgress. 17 = registrationComplete. 18 = accessDenied. 19 = operational. 20 = bpiInit. 21 = forwardingDisabled.
expiryDeadline	TimeStamp	0..1	Time stamp.
NOTE 1: At least one of callbackReference and websockNotifConfig shall be provided by the service consumer. If both are provided, it is up to FAIS to choose an alternative and return only that alternative in the response, as described in ETSI GS MEC 009 [i.1], clause 6.12a. NOTE 2: 'notReady' indicates that the CM has not started the registration process yet. 'notSynchronized' indicates that the CM has not initiated or completed the synchronization of the downstream physical layer. 'phySynchronized' indicates that the CM has completed the synchronization of the downstream physical layer. 'dsTopologyResolutionInProgress' indicates that the CM is attempting to determine its MD-DS-SG. 'usParametersAcquired' indicates that the CM has completed the upstream parameters acquisition or have completed the downstream and upstream service groups resolution. 'rangingInProgress' indicates that the CM has initiated the initial ranging process. 'rangingComplete' indicates that the CM has completed initial ranging and received a Ranging Status of success from the CMTS in the RNG-RSP message. 'eaeInProgress' indicates that the CM has sent an Auth Info message for EAE. 'dhcpv4InProgress' indicates that the CM has sent a DHCPv4 DISCOVER to gain IP connectivity. 'dhcpv6InProgress' indicates that the CM has sent a DHCPv6 Solicit message. 'dhcpv4Complete' indicates that the CM has received a DHCPv4 ACK message from the CMTS. 'dhcpv6Complete' indicates that the CM has received a DHCPv6 Reply message from the CMTS. 'todEstablished' indicates that the CM has successfully acquired time of day. 'securityEstablished' indicates that the CM has successfully completed the BPI initialization process. 'configFileDownloadComplete' indicates that the CM has completed the config file download process. 'registrationInProgress' indicates that the CM has sent a Registration Request. 'registrationComplete' indicates that the CM has successfully completed the Registration process with the CMTS. 'accessDenied' indicates that the CM has received a registration aborted notification from the CMTS. 'operational' indicates that the CM has completed all necessary initialization steps and is operational. 'bpiInit' indicates that the CM has started the BPI initialization process as indicated in the CM config file. 'forwardingDisabled' indicates that the registration process was completed, but the network access option in the received configuration file prohibits forwarding.			

6.3.5 Type: SubscriptionLinkList

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

Table 6.3.5-1: Attributes of the SubscriptionLinkList

Attribute name	Data type	Cardinality	Description
_links	Structure (inlined)	1	Hyperlinks related to the resource.
>self	LinkType	1	URI of this resource.
>subscriptions	Structure (inlined)	0..N	The service consumer'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 6.3.2, 6.3.3, 6.3.4 and 6.3.6: • "OnuAlarmSubscription" • "DevInfoSubscription" • "CmConnSubscription" • "AniAlarmSubscription"

6.3.6 Type: AniAlarmSubscription

This type represents a subscription to ANI alarm notifications from FAIS.

The attributes of the AniAlarmSubscription shall follow the indications provided in table 6.3.6-1, as defined in [i.7].

Table 6.3.6-1: Attributes of the AniAlarmSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "AniAlarmSubscription".
callbackReference	Uri	0..1	URI exposed by the client on which to receive notifications via HTTP. See note.
requestTestNotification	Boolean	0..1	Shall be set to TRUE by the service consumer to request a test notification via HTTP on the callbackReference URI, as described in ETSI GS MEC 009 [i.1], clause 6.12a. Default: FALSE.
websockNotifConfig	WebsockNotifConfig	0..1	Provides details to negotiate and signal the use of a Websocket connection between FAIS and the service consumer for notifications. See note.
_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 FAI API as it acts as an ID for the subscription.
filterCriteriaAniAlarm	Structure (inlined)	1	List of filtering criteria for the subscription. Any filtering criteria from below, which is included in the request, shall also be included in the response.
>customerPremisesInfo	CpInfo	0..N	0 to N physical locations of the customer sites.
>onuId	String	1..N	1 to N unique identifiers for the optical network units. Typical it is the unique serial number for each ONU, which contains the vendor ID and version number. The first four bytes are an ASCII-encoded four-letter vendor ID. The second four bytes are a binary encoded serial number, under the control of the ONU vendor.
>aniId	Structure (inlined)	1..N	
>>onuId	String	1	The unique identifiers for the optical network unit,
>>aniIndex	String	1	The index of an access network interface supported by the optical network unit.

Attribute name	Data type	Cardinality	Description
>alarms	Enum	0..N	In case alarms is not included in the subscription request, the default value -1 = All shall be used and included in the response: 0 = LowReceivedOpticalPower. (Received downstream optical power below threshold.) 1 = HighReceivedOpticalPower. (Received downstream optical power above threshold.) 2 = SignalFalure. (Bit error-based signal fail.) 3 = SignalDegrade. (Bit error-based signal degrade.) 4 = LowTransmitOpticalPower. (Transmit optical power below lower threshold.) 5 = HighTransmitOpticalPower. (Transmit optical power above upper threshold.) 6 = LaserBiasCurrent. (Laser bias current above threshold determined by vendor.) -1 = All.
expiryDeadline	TimeStamp	0..1	Time stamp.
NOTE: At least one of callbackReference and websockNotifConfig shall be provided by the service consumer. If both are provided, it is up to FAIS to choose an alternative and return only that alternative in the response, as described in ETSI GS MEC 009 [i.1], clause 6.12a.			

6.4 Notification data types

6.4.1 Introduction

This clause defines data structures that define notifications.

6.4.2 Type: OnuAlarmNotification

This type represents a notification from FAIS with regards to ONU alarms. The Notification is sent by the FAIS to inform about the alarms of an optical network unit.

The attributes of the OnuAlarmNotification shall follow the indications provided in table 6.4.2-1.

Table 6.4.2-1: Attributes of the OnuAlarmNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "OnuAlarmNotification".
timeStamp	TimeStamp	0..1	Time stamp.
customerPremisesInfo	CpInfo	0..N	The physical location of the related customer sites.
onuId	String	1	The unique identifier for an optical network unit.
alarm	Enum	1	Indicates the alarm of the ONU: 0 = EquipmentAlarm. 1 = PoweringAlarm. 2 = BatteryMissing. 3 = BatteryFailure. 4 = BatteryLow. 5 = PhysicalIntrusion. 6 = OnuSelfTestFailure. 7 = DyingGasp. 8 = TemperatureYellow. 9 = TemperatureRed. 10 = VoltageYellow. 11 = VoltageRed. 12 = OnuManualPowerOff. 13 = InvImage. 14 = PseOverloadYellow. 15 = PseOverloadRed.

6.4.3 Type: DevInfoNotification

This type represents a notification from FAIS with regards to device information. The Notification is sent by the FAIS to inform about the abnormal operational status of a device.

The attributes of the DevInfoNotification shall follow the indications provided in table 6.4.3-1.

Table 6.4.3-1: Attributes of the DevInfoNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "DevInfoNotification".
timeStamp	TimeStamp	0..1	Time stamp.
deviceId	String	1	The identifier for a device.
deviceErrStatus	Enum	1	Indicates the abnormal operational status of the device. Enumeration of: 1 = Error. 2 = Disabled.

6.4.4 Type: CmConnNotification

This type represents a notification from FAIS with regards to cable modem connectivity state.

The attributes of the CmConnNotification shall follow the indications provided in table 6.4.4-1.

Table 6.4.4-1: Attributes of the CmConnNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "CmconnNotification".
timeStamp	TimeStamp	0..1	Time stamp.
customerPremisesInfo	CpInfo	0..1	The physical location of a customer site.
cmIf	Structure (inlined)	1..N	
>cmId	String	1	The Cable Modem identifier.
>ifIndex	String	1..N	The MAC Domain interface index of the CM.
>cmRegState	Enum	1..N	It indicates the CM connectivity state. Enumeration of: 1 = notReady. 2 = notSynchronized. 3 = phySynchronized. 4 = dsTopologyResolutionInProgress. 5 = usParametersAcquired. 6 = rangingInProgress. 7 = rangingComplete. 8 = eaeInProgress. 9 = dhcpv4InProgress. 10 = dhcpv6InProgress. 11 = dhcpV4Complete. 12 = dhcpV6Complete. 13 = todEstablished. 14 = securityEstablished. 15 = configFileDownloadComplete. 16 = registrationInProgress. 17 = registrationComplete. 18 = accessDenied. 19 = operational. 20 = bpiInit. 21 = forwardingDisabled.

6.4.5 Type: ExpiryNotification

This type represents a notification from FAIS with regards to expiry of the existing subscription.

The Notification is sent by the FAIS to send information about expiry of a subscription.

Table 6.4.5-1: Attributes of the ExpiryNotification

Attribute name	Data type	Cardinality	Description
timeStamp	TimeStamp	0..1	Time stamp.
_links	Structure (inlined)	1	List of hyperlinks related to the resource.
>self	Uri	1	Self-referring URI. This shall be included in the response from the FAIS. The URI shall be unique within the FAI API as it acts as an ID for the subscription.
expiryDeadline	TimeStamp	1	Time stamp.

6.4.6 Type: AniAlarmNotification

This type represents a notification from FAIS with regards to ANI alarms. The Notification is sent by the FAIS to inform about the alarms of an optical network unit.

The attributes of the AniAlarmNotification shall follow the indications provided in table 6.4.6-1.

Table 6.4.6-1: Attributes of the AniAlarmNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "AniAlarmNotification".
timeStamp	TimeStamp	0..1	Time stamp.
customerPremisesInfo	CpInfo	0..N	The physical location of the related customer sites.
onuId	String	0..1	The unique identifier for an optical network unit.
aniId	Structure (inlined)	0..1	
>onuId	String	1	The unique identifiers for the optical network unit.
>aniIndex	String	1	The index of an access network interface supported by the optical network unit.
alarm	Enum	1	Indicates the alarm of the ANI: 0 = LowReceivedOpticalPower. 1 = HighReceivedOpticalPower. 2 = SignalFailure. 3 = SignalDegrade. 4 = LowTransmitOpticalPower. 5 = HighTransmitOpticalPower. 6 = LaserBiasCurrent.

6.4.7 Type: TestNotification

This type represents a test notification from a FAIS to determine if the Websocket method is to be utilized to issue notifications for a subscription, as defined in ETSI GS MEC 009 [i.1], clause 6.12a.

Table 6.4.7-1: Attributes of the TestNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "TestNotification".
_links	Structure (inlined)	1	Hyperlink related to the resource.
>subscription	LinkType	1	URI identifying the subscription for the test notification.

6.5 Referenced structured data types

6.5.1 Introduction

This clause defines data structures that are referenced from data structures defined in the previous clauses, but are neither resource representations nor bound to any pub/sub mechanism.

6.5.2 Type: TimeStamp

This type represents a time stamp.

Table 6.5.2-1: Attributes of the 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.

6.5.3 Type: CpInfo

This type represents a customer premises (physical location). In order to provide the accurate location information either (latitude, longitude) or postal code needs to be present.

Table 6.5.3-1: Attributes of the CpInfo

Attribute name	Data type	Cardinality	Description
latitude	Float	0..1	Latitude (DATUM=WGS84) -90 to 90 in decimal degree format DDD.ddd
longitude	Float	0..1	Longitude (DATUM=WGS84) -180 to 180 in decimal degree format DDD.ddd
postalCode	String	0..1	Postal code for the location

6.5.4 Type: IPPingDiagnostics

This type represents an IP-layer ping test.

The attributes of the IPPingDiagnostics shall follow the notations provided in table 6.5.4-1, as defined in [i.5].

Table 6.5.4-1: Attributes of the IPPingDiagnostics

Attribute name	Data type	Cardinality	Description
diagnosticsState	Enum	1	The state of the IP ping test. Enumeration of: 1 = None. 2 = Requested. 3 = Complete. 4 = Error_CannotResolveHostName. 5 = Error_Internal. 6 = Error_Other.
host	String	0..1	Host name or address of the host to ping.
numberOfRepetitions	Integer	0..1	Number of repetitions of the ping test to perform before reporting the results.
timeout	Integer	0..1	Timeout in milliseconds for the ping test.
dataBlockSize	Integer	0..1	Size of the data block in bytes to be sent for each ping.
successCount	Integer	0..1	Result parameter indicating the number of successful pings (those in which a successful response was received prior to the timeout) in the most recent ping test.
failureCount	Integer	0..1	Result parameter indicating the number of failed pings in the most recent ping test.
averageResponseTime	Integer	0..1	Result parameter indicating the average response time in milliseconds over all repetitions with successful responses of the most recent ping test. If there were no successful responses, this value shall be zero.
minimumResponseTime	Integer	0..1	Result parameter indicating the minimum response time in milliseconds over all repetitions with successful responses of the most recent ping test. If there were no successful responses, this value shall be zero.
maximumResponseTime	Integer	0..1	Result parameter indicating the maximum response time in milliseconds over all repetitions with successful responses of the most recent ping test. If there were no successful responses, this value shall be zero.

6.5.5 Type: TraceRouteDiagnostics

This type represents an IP-layer trace-route test.

The attributes of the TraceRouteDiagnostics shall follow the notations provided in table 6.5.5-1, as defined in [i.5].

Table 6.5.5-1: Attributes of the TraceRouteDiagnostics

Attribute name	Data type	Cardinality	Description
diagnosticsState	Enum	1	The state of the IP trace-route test. Enumeration of: 1 = None. 2 = Requested. 3 = Complete. 4 = Error_CannotResolveHostName. 5 = Error_MaxHopCountExceeded. 6 = Error_Internal. 7 = Error_Other.
host	String	0..1	Host name or address of the host to find a route to.
timeout	Integer	0..1	Timeout in milliseconds for the trace route test.
dataBlockSize	Integer	0..1	Size of the data block in bytes to be sent for each trace route.
maxHopCount	Integer	0..1	The maximum number of hop used in outgoing probe packets. The default is 30 hops.
responseTime	Integer	0..1	Result parameter indicating the response time in milliseconds the most recent trace route test. If a route could not be determined, this value shall be zero.
numberOfRouteHops	Integer	0..1	Result parameter indicating the number of hops within the discovered route. If a route could not be determined, this value shall be zero.

6.5.6 Type: DownloadDiagnostics

This type represents a HTTP and FTP DownloadDiagnostics test.

The attributes of the DownloadDiagnostics shall follow the notations provided in table 6.5.6-1, as defined in [i.5].

Table 6.5.6-1: Attributes of the DownloadDiagnostics

Attribute name	Data type	Cardinality	Description
diagnosticsState	Enum	1	The state of the HTTP and FTP download test. Enumeration of: 1 = None. 2 = Requested. 3 = Completed. 4 = Error_InitConnectionFailed. 5 = Error_NoResponse. 6 = Error_TransferFailed. 7 = Error_PasswordRequestFailed. 8 = Error_LoginFailed. 9 = Error_NoTransferMode. 10 = Error_NoPASV. 11 = Error_IncorrectSize. 12 = Error_Timeout.
downloadURI	Uri	0..1	The URI for the device to perform the download on.
rOMTime	TimeStamp	0..1	Request time in UTC, which shall be specified to microsecond precision.
bOMTime	TimeStamp	0..1	Begin of transmission time in UTC, which shall be specified to microsecond precision.
eOMTime	TimeStamp	0..1	End of transmission in UTC, which shall be specified to microsecond precision.

Attribute name	Data type	Cardinality	Description
testBytesReceived	Integer	0..1	The test traffic received in bytes during the FTP/HTTP transaction including FTP/HTTP headers, between *bOMTime* and *eOMTime*.
totalBytesReceived	Integer	0..1	The total number of bytes received on the Interface between b*OMTime* and *eOMTime*.
tCPOpenRequestTime	TimeStamp	0..1	Request time in UTC, which shall be specified to microsecond precision.
tCPOpenReponseTime	TimeStamp	0..1	Response time in UTC, which shall be specified to microsecond precision.

6.5.7 Type: UploadDiagnostics

This type represents a HTTP and FTP UploadDiagnostics test.

The attributes of the UploadDiagnostics shall follow the notations provided in table 6.5.7-1, as defined in [i.5].

Table 6.5.7-1: Attributes of the UploadDiagnostics

Attribute name	Data type	Cardinality	Description
diagnosticsState	Enum	1	The state of the HTTP and FTP download test. Enumeration of: 1 = None. 2 = Requested. 3 = Completed. 4 = Error_InitConnectionFailed. 5 = Error_NoResponse. 6 = Error_PasswordRequestFailed. 7 = Error_LoginFailed. 8 = Error_NoTransferMode. 9 = Error_NoPASV. 10 = Error_NoCWD. 11 = Error_NoSTOR. 12 = Error_NoTransferComplete.
uploadURI	Uri	0..1	The URI for the device to perform the upload to.
testFileLength	Integer	0..1	The size of the file (in bytes) to be uploaded to the server.
rOMTime	TimeStamp	0..1	Request time in UTC, which shall be specified to microsecond precision.
bOMTime	TimeStamp	0..1	Begin of transmission time in UTC, which shall be specified to microsecond precision.
eOMTime	TimeStamp	0..1	End of transmission in UTC, which shall be specified to microsecond precision.
testBytesSent	Integer	0..1	The test traffic sent in bytes during the FTP/HTTP transaction including FTP/HTTP headers, between *bOMTime* and *eOMTime*.
tCPOpenRequestTime	TimeStamp	0..1	Request time in UTC, which shall be specified to microsecond precision.
tCPOpenReponseTime	TimeStamp	0..1	Response time in UTC, which shall be specified to microsecond precision.

6.5.8 Type: LinkType

This type represents a type of link.

Table 6.5.8-1: Attributes of the LinkType

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

6.5.9 Type: WebsockNotifConfig

This type represents configuration for the delivery of subscription notifications over Websockets per the pattern defined in ETSI GS MEC 009 [i.1] clause 6.12a.

Table 6.5.9-1: Attributes of the WebsockNotifConfig

Attribute name	Data type	Cardinality	Description
websocketUri	Uri	0..1	Set by FAIS to indicate to the service consumer the Websocket URI to be used for delivering notifications.
requestWebsocketUri	Boolean	0..1	Set to true by the service consumer to indicate that Websocket delivery is requested.

7 API definitions

7.1 Introduction

This clause defines the resources and operations of the FAI 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 ("http" or "https"), host and optional port, and an optional prefix string. The "apiName" shall be set to "fai" and "apiVersion" shall be set to "v1" for the current version of the specification. The API shall support HTTP over TLS (also known as HTTPS) using TLS version 1.2 as defined by IETF RFC 5246 [5]. 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. All resource URIs in the clauses below are defined relative to the above root URI.

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 [3] with bearer tokens according to IETF RFC 6750 [4]. See clause 6.16 of ETSI GS MEC 009 [i.1] for more information. The token endpoint can be discovered as part of the service availability query procedure defined in ETSI GS MEC 011 [i.2]. How the client credentials are provisioned into the MEC application 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 [i.1] 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.

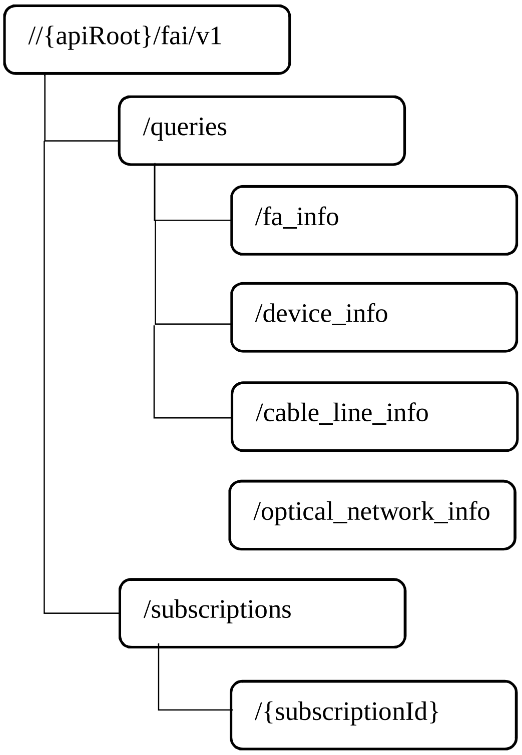

Figure 7.2-1: Resource URI structure of the FAI API

Table 7.2-1: Resources and methods overview

Resource name	Resource URI	HTTP method	Meaning
fixed access information	/queries/fa_info	GET	Retrieve current status of the FAI.
device information	/queries/device_info	GET	Retrieve current status of the device information.
cable line information	/queries/cable_line_info	GET	Retrieve current status of the cable line information.
optical network information	/queries/optical_network_info	GET	Retrieve current status of the optical network information.
all subscriptions for a subscriber	/subscriptions	GET	Retrieve a list of active subscriptions for this subscriber.
		POST	Create a new subscription.
existing subscription	/subscriptions/{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.

7.3 Resource: Resource: fa_info

7.3.1 Description

This resource is queried to retrieve information on the available fixed access networks.

7.3.2 Resource definition

Resource URI: **{apiRoot}/fai/v1/queries/fa_info**

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 "fa_info"

Name	Definition
apiRoot	See clause 7.2.

7.3.3 Resource methods

7.3.3.1 GET

The GET method is used to query information about the available fixed access networks.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the 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
customerPremisesInfo	CpInfo	0..N	Comma separated list of customer premises information defined in clause 6.2.2.
lastMileTech	Integer	0..N	Comma separated list of last mile technologies defined in clause 6.2.2.
interfaceType	Integer	0..N	Comma separated list of interface types defined in clause 6.2.2.
dsbw	Integer	0..N	Comma separated list of the bandwidth (in Mbps) from the network towards the customer site defined in clause 6.2.2.
usbw	Integer	0..N	Comma separated list of the bandwidth (in Mbps) from the customer site towards the network defined in clause 6.2.2.
latency	Integer	0..N	Comma separated list of the maximum baseline latency (in ms) between customer site and service edge node defined in clause 6.2.2.

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
	FaInfo	0..N	200 OK	Upon success, a response body containing the information of zero or more fix access networks is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed in the request. This error condition can also occur if the target area for the request is considered too large. 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.

Response body	Data type	Cardinality	Response Codes	Remarks
	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 the 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: device_info

7.4.1 Description

This resource is queried to retrieve information on the devices that are connected to a fixed access network.

7.4.2 Resource definition

Resource URI: **{apiRoot}/fai/v1/queries/device_info**

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 "device_info"

Name	Definition
apiRoot	See clause 7.2.

7.4.3 Resource methods

7.4.3.1 GET

The GET method is used to query information about the devices that are connected to a fixed access network.

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

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

Name	Data type	Cardinality	Remarks
gwId	String	0..N	Comma separated list of gateway identifier defined in clause 6.2.3.
deviceId	String	0..N	Comma separated list of device identifier defined in clause 6.2.3.
deviceStatus	Integer	0..N	Comma separated list of device status defined in clause 6.2.3.

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
	DeviceInfo	0..N	200 OK	Upon success, a response body containing the information of zero or more devices is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed in the request. This error condition can also occur if the target area for the request is considered too large. 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 the 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

Not applicable.

7.4.3.5 DELETE

Not applicable.

7.5 Resource: cable_line_info

7.5.1 Description

This resource is queried to retrieve information of the cable line of a fixed access network.

7.5.2 Resource definition

Resource URI: **{apiRoot}/fai/v1/queries/cable_line_info**

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 "cable_line_info"

Name	Definition
apiRoot	See clause 7.2.

7.5.3 Resource methods

7.5.3.1 GET

The GET method is used to query information about the cable line of a fixed access network.

This method shall support the URI query parameters, 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
customerPremisesInfo	CpInfo	0..N	Comma separated list of customer premises information defined in clause 6.2.4.
cmId	String	0..N	Comma separated list of Cable Modem identifiers defined in clause 6.2.4.

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
	CableLineInfo	0..N	200 OK	Upon success, a response body containing the information of zero or more cable lines is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed in the request. This error condition can also occur if the target area for the request is considered too large. 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 the 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

Not applicable.

7.5.3.3 PATCH

Not applicable.

7.5.3.4 POST

Not applicable.

7.5.3.5 DELETE

Not applicable.

7.6 Resource: optical_network_info

7.6.1 Description

This resource is queried to retrieve information of an optical network.

7.6.2 Resource definition

Resource URI: **{apiRoot}/fai/v1/queries/optical_network_info**

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 "optical_network_info"

Name	Definition
apiRoot	See clause 7.2.

7.6.3 Resource methods

7.6.3.1 GET

The GET method is used to query information about the optical network.

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
customerPremisesInfo	CpInfo	0..N	Comma separated list of customer premises information defined in clause 6.2.5.
ponYS_ID	String	0..N	Comma separated list of optical system identifiers defined in clause 6.2.5.
onuId	String	0..N	Comma separated list of optical network unit identifiers defined in clause 6.2.5.

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
	PonInfo	0..N	200 OK	Upon success, a response body containing the information of zero or more optical networks is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed in the request. This error condition can also occur if the target area for the request is considered too large. 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 the 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: subscriptions

7.7.1 Description

This resource contains various resources related to subscriptions for notifications.

7.7.2 Resource definition

Resource URI: **{apiRoot}/fai/v1/subscriptions**

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 "subscriptions"

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 for this requestor. Upon success, the response contains entity body with the list of links to the subscriptions that are present for the requestor.

This method shall support the URI query parameters, 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 parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
subscription_type	String	0..1	Query parameter to filter on a specific subscription type. Permitted values: - onu_alarm Optical Network Unit alarm - device_abnormal_alert Device abnormal alert - cm_connectivity_state Cable Modem connectivity state - ani_alarm Access Node Interface alarm

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 subscriptions is returned. The list shall only contain subscriptions of the subscription type specified by the URI query parameters.
	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.

Response body	Data type	Cardinality	Response Codes	Remarks
	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 the 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 FAI 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

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>{NotificationSubscription}</td><td>1</td><td colspan="2">The entity body in the request contains data type of the specific FAI event subscription that is to be created, where the data type options are listed below:
• OnuAlarmSubscription.
• DevInfoSubscription.
• CmConnSubscription.
• AniAlarmSubscription.</td></tr>
<tr><th rowspan="8">Response body</th><th>Data type</th><th>Cardinality</th><th>Response Codes</th><th>Remarks</th></tr>
<tr><td>{NotificationSubscription}</td><td>1</td><td>201 Created</td><td>Indicates successful resource creation, where the resource URI shall be returned in the HTTP Location header field.
In the returned NotificationSubscription structure, the created subscription is described using the appropriate data type from the list below:
• OnuAlarmSubscription.
• DevInfoSubscription.
• CmConnSubscription.
• AniAlarmSubscription.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>400 Bad Request</td><td>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.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>401 Unauthorized</td><td>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.</td></tr>
<tr><td>ProblemDetails</td><td>1</td><td>403 Forbidden</td><td>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.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>404 Not Found</td><td>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.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>406 Not Acceptable</td><td>It is used to indicate that the server cannot provide the any of the content formats supported by the client.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>415 Unsupported Media Type</td><td>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.</td></tr>
</table>

Response body	Data type	Cardinality	Response Codes	Remarks
	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 an JSON request body is syntactically correct but semantically incorrect, for example if the target area for the request is considered too large. 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: existing subscription

7.8.1 Description

This resource represents a subscription that the client has created to receive FAI event notifications.

7.8.2 Resource definition

Resource URI: **{apiRoot}/fai/v1/subscriptions/{subscriptionId}**

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 "existing subscription"

Name	Definition
apiRoot	See clause 7.2.
subscriptionId	Refers to created subscription, where the FAI API allocates a unique resource name for this subscription. The resource name can be also used to identify the resource.

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 URI query parameters, 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**
	{NotificationSubscription}	1	200 OK	Upon success, a response body containing data type describing the specific FAI event subscription is returned. The allowed data types for subscriptions are as follows: • OnuAlarmSubscription. • DevInfoSubscription. • CmConnSubscription. • AniAlarmSubscription.
	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 the 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

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>{NotificationSubscription}</td><td>1</td><td colspan="2">New NotificationSubscription is included as entity body of the request. The allowed data types for subscriptions are as follows:
• OnuAlarmSubscription.
• DevInfoSubscription.
• CmConnSubscription.
• AniAlarmSubscription.</td></tr>
<tr><th rowspan="8">Response body</th><th>Data type</th><th>Cardinality</th><th>Response Codes</th><th>Remarks</th></tr>
<tr><td>{NotificationSubscription}</td><td>1</td><td>200 OK</td><td>Upon success, a response body containing data type describing the updated subscription is returned. The allowed data types for subscriptions are as follows:
• OnuAlarmSubscription.
• DevInfoSubscription.
• CmConnSubscription.
• AniAlarmSubscription.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>400 Bad Request</td><td>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.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>401 Unauthorized</td><td>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.</td></tr>
<tr><td>ProblemDetails</td><td>1</td><td>403 Forbidden</td><td>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.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>404 Not Found</td><td>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.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>406 Not Acceptable</td><td>It is used to indicate that the server cannot provide the any of the content formats supported by the client.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>412 Precondition Failed</td><td>It is used when a condition has failed during conditional requests, e.g. when using ETags to avoid write conflicts when using PUT.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
</table>

Response body	Data type	Cardinality	Response Codes	Remarks
	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 an JSON request body is syntactically correct but semantically incorrect, for example if the target area for the request is considered too large. 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.

7.8.3.3 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 subscription.

This method shall support the URI query parameters, 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

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>n/a</td><td></td><td colspan="2"></td></tr>
<tr><th rowspan="6">Response body</th><th>Data type</th><th>Cardinality</th><th>Response Codes</th><th>Remarks</th></tr>
<tr><td>n/a</td><td></td><td>204 No Content</td><td>Upon success, a response 204 No Content without any response body is returned.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>401 Unauthorized</td><td>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.</td></tr>
<tr><td>ProblemDetails</td><td>1</td><td>403 Forbidden</td><td>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.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>404 Not Found</td><td>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.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>429 Too Many Requests</td><td>It is used when a rate limiter has triggered.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
</table>

Annex A (informative): Mapping of permissions for RESTful API and topic based alternative transport

A.1 Overview

This annex provides mappings of permissions for topics between RESTful API and topic based alternative transport. ETSI GS MEC 009 [i.1] describes how permissions for topics between RESTful API and alternative transport can be mapped with each other. This annex uses the template for permissions mapping as defined in that group specification.

A.2 Mapping of permissions - RESTful and topic based alternative transport

Table A.2-1 lists the permission categories for each topic currently included in FAI API specification.

Table A.2-1: Definition of permissions for FAIS

Permission identifier	Display name	Remarks
fa_info	FA Info	Query
device_info	Device Info	Query
cable_line_info	Cable Line Info	Query
optical_network_info	Optical Network Info	Query
onu_alarm	ONU Alarm	Subscribe-Notify
device_abnormal_alert	Device Abnormal Alert	Subscribe-Notify
cm_connectivity_state	CM Connectivity State	Subscribe-Notify
ani_alarm	ANI Alarm	Subscribe-Notify

Table A.2-2 describes how permission identifiers can be mapped to resources in the FAI RESTful API as defined in the present document.

Table A.2-2: Permission identifiers mapping for transport "REST"

Permission identifier	Specification
fa_info	Resource: …/fai/v1/queries/fa_info
device_info	Resource: …/fai/v1/queries/device_info
cable_line_info	Resource: …/fai/v1/queries/cable_line_info
optical_network_info	Resource: …/fai/v1/queries/optical_network_info
onu_alarm	Resource: …/fai/v1/subscriptions/onu_alarm
device_abnormal_alert	Resource: …/fai/v1/subscriptions/device_abnormal_alert
cm_connectivity_state	Resource: …/fai/v1/subscriptions/cm_connectivity_state
ani_alarm	Resource: …/fai/v1/subscriptions/ani_alarm

Table A.2-3 describes how the permission identifiers can be mapped to topics offered over topic-based message bus.

Table A.2-3: Permission identifiers mapping for transport "Topic-based message bus"

Permission identifier	Specification
fa_info	Topic: /fai/fa_Info
device_info	Topic: /fai/device_Info
cable_line_info	Topic: /fai/cable_line_info
optical_network_info	Topic: /fai/optical_network_info
onu_alarm	Topic: /fai/onu_alarm
device_abnormal_alert	Topic: /fai/device_abnormal_alert
cm_connectivity_state	Topic: /fai/cm_connectivity_state
ani_alarm	Topic: /fai/ani_alarm

Annex B (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 FAI API a supplementary description file compliant to the OpenAPI Specification [i.12].

In addition, a further supplementary file defining the data types in protocol buffers format, as defined in the Protocol Buffers Language Specification [i.13], is provided.

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

The supplementary files, relating to the present document, are located at https://forge.etsi.org/rep/mec/gs029-fai-api.

History

Document history		
V2.1.1	July 2019	Publication
V2.2.1	January 2022	Publication