ETSI GS MEC 048 V3.1.1 (2024-04)

Multi-access Edge Computing (MEC); Enablement API for Customer Self-Service

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
DGS/MEC-0048v311SelfServEnabl

Keywords
API, MEC

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:
https://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

If you find a security vulnerability in the present document, please report it through our
Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

Notice of disclaimer & limitation of liability

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

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

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.
The content of the PDF version shall not be modified without the written authorization of ETSI.
The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2024.
All rights reserved.

Contents

Intellectual Property Rights 6
Foreword 6
Modal verbs terminology 6
1 Scope 7
2 References 7
2.1 Normative references 7
2.2 Informative references 7
3 Definition of terms, symbols and abbreviations 8
3.1 Terms 8
3.2 Symbols 8
3.3 Abbreviations 8
4 Overview 8
5 Description of the services (informative) 8
5.1 Introduction 8
5.2 Sequence diagrams 9
5.2.1 General 9
5.2.2 Tenant management 9
5.2.2.1 Introduction 9
5.2.2.2 Tenant resource creation 9
5.2.2.3 Tenant resource deletion 10
5.2.2.4 Tenant resource update 10
5.2.2.5 Tenant resource query 10
5.2.3 Computing resource management 11
5.2.3.1 Introduction 11
5.2.3.2 Computing resource request (per-tenant) 12
5.2.3.3 Computing resource request update (per-tenant) 12
5.2.3.4 Computing resource quota query (per tenant) 13
5.2.4 REST based subscribe-notify model 14
5.2.4.1 Subscribing to CSE event notifications 14
5.2.4.2 Receiving notification on expiry of CSE event subscription 14
5.2.4.3 Updating subscription for CSE event notifications 15
5.2.4.4 Unsubscribing from CSE event notifications 15
5.2.5 Receiving CSE event notifications about resource usage in a MEC system 16
5.2.6 Receiving CSE event notifications about resource usage in edge sites 16
6 Data model 17
6.1 Introduction 17
6.2 Resource data types 17
6.2.1 Overview 17
6.2.2 Type: TenantInfo 17
6.2.3 Type: ResourceQuotaInfo 18
6.2.4 Type: SiteResourceQuotaInfo 18
6.3 Subscription data types 19
6.3.1 Introduction 19
6.3.2 Type: ResourceUsageSubscription 19
6.3.3 Type: SiteResourceUsageSubscription 19
6.3.4 Type: SubscriptionLinkList 20
6.4 Notifications data types 21
6.4.1 Introduction 21
6.4.2 Type: ResourceUsageNotification 21
6.4.3 Type: SiteResourceUsageNotification 21
6.5 Referenced structured data types 22
6.5.1 Introduction 22
6.5.2 Type: SiteInfo 22

6.5.3 Type: ResourceInfo 22
7 API definitions 23
7.1 Introduction 23
7.2 Global definitions and resource structure 23
7.3 Resource: a list of tenants 24
7.3.1 Description 24
7.3.2 Resource definition 24
7.3.3 Resource methods 25
7.3.3.1 GET 25
7.3.3.2 PUT 25
7.3.3.3 PATCH 25
7.3.3.4 POST 25
7.3.3.5 DELETE 26
7.4 Resource: individual tenant 26
7.4.1 Description 26
7.4.2 Resource definition 26
7.4.3 Resource methods 27
7.4.3.1 GET 27
7.4.3.2 PUT 27
7.4.3.3 PATCH 28
7.4.3.4 POST 28
7.4.3.5 DELETE 28
7.5 Resource: per system resource quota 29
7.5.1 Description 29
7.5.2 Resource definition 29
7.5.3 Resource methods 29
7.5.3.1 GET 29
7.5.3.2 PUT 30
7.5.3.3 PATCH 31
7.5.3.4 POST 31
7.5.3.5 DELETE 32
7.6 Resource: a list of per site resource quota 32
7.6.1 Description 32
7.6.2 Resource definition 32
7.6.3 Resource methods 32
7.6.3.1 GET 32
7.6.3.2 PUT 33
7.6.3.3 PATCH 33
7.6.3.4 POST 33
7.6.3.5 DELETE 34
7.7 Resource: individual per site resource quota 34
7.7.1 Description 34
7.7.2 Resource definition 34
7.7.3 Resource methods 34
7.7.3.1 GET 34
7.7.3.2 PUT 35
7.7.3.3 PATCH 36
7.7.3.4 POST 36
7.7.3.5 DELETE 36
7.8 Resource: subscriptions 36
7.8.1 Description 36
7.8.2 Resource definition 36
7.8.3 Resource methods 37
7.8.3.1 GET 37
7.8.3.2 PUT 38
7.8.3.3 PATCH 38
7.8.3.4 POST 38
7.8.3.5 DELETE 39
7.9 Resource: existing subscription 39
7.9.1 Description 39
7.9.2 Resource definition 39

7.9.3 Resource methods 40
7.9.3.1 GET 40
7.9.3.2 PUT 41
7.9.3.3 PATCH 42
7.9.3.4 POST 42
7.9.3.5 DELETE 42
8 OpenAPI definitions (informative) 43
8.1 Introduction 43
8.2 Definition and operation on Resources 43
Annex A (informative): Complementary material for API utilization 58
Annex B (informative): Change History 59
History 60

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 specifies the enablement APIs produced by MEO over Mm1 reference point to support customer self-service portal. This includes the related aspects on tenant management, per tenant application management, per tenant resource management, basic monitoring per tenant, per tenant accounting capabilities. It describes the information flows, required information, and specifies the necessary operations, data models and API definitions with example codes.

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

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 GR MEC 001: "Multi-access Edge Computing (MEC); Terminology".

[i.2] IETF RFC 4122: "A Universally Unique IDentifier (UUID) URN Namespace".

[i.3] OpenAPI™ Specification.

[i.4] IETF RFC 7807: "Problem Details for HTTP APIs".

[i.5] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GR MEC 001 [i.1] and the following apply:

tenant: user who shares the access to the resources (e.g. a set of physical, virtual or service resources) in a private or public environment that is isolated from other users

NOTE: A tenant in MEC can be associated with an enterprise customer who has an account with the MEC system provider. Such enterprise customer may be associated with one or multiple tenants in a MEC system.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GR MEC 001 [i.1] and the following apply:

CSE Customer Self-service Enablement

4 Overview

The present document specifies the Customer Self-service Enablement (CSE) APIs over Mm1 reference point in order to support customer self-service portal.

Clause 5 introduces the functionalities enabled by CSE APIs and how the CSE services may be used by a service consumer. It provides the high-level information flows and describes the necessary operations.

The information that can be exchanged over the CSE APIs is described in clause 6 which defines detailed data models with description on all information elements.

Clause 7 specifies the CSE APIs providing detailed information how information elements are mapped into a RESTful API design.

The informative OpenAPI definitions are provided in clause 8 to illustrate the usage of the defined data model.

5 Description of the services (informative)

5.1 Introduction

A self-service portal is the customer facing part of the IT service management tool. In the context of MEC, self-service portals provide customers quick and easy access to the various services and resources offered by the MEC system. The CSE services are intended to support customer self-service portal, enabling simple application management and basic tenant, resource management capabilities.

Via Mm1 reference point the CSE APIs enable the basic functions, including:

- Tenant management:
  - creation of tenant resource;
  - deletion of tenant resource;

  - updating tenant resource; and
  - querying tenant resource.
- Computing resource management:
  - per tenant resource request in an edge site or in a MEC system;
  - per tenant resource request update in an edge site or in a MEC system; and
  - per tenant resource quota query in an edge site or in a MEC system.
- REST based subscribe-notify model:
  - subscribing to CSE event notifications;
  - receiving notifications on subscribed CSE events;
  - updating subscription for CSE event notifications; and
  - unsubscribing from CSE event notifications.

5.2 Sequence diagrams

5.2.1 General

The following clauses describe how the CSE services can be used by a service consumer via Mm1 reference point. The related sequence diagrams are presented.

5.2.2 Tenant management

5.2.2.1 Introduction

In the context of the present document, a tenant typically is associated with an enterprise customer who has an account with the MEC system provider. The basic functionalities for tenant management include:

- tenant resource creation;
- tenant resource deletion;
- tenant resource update; and
- tenant resource query.

5.2.2.2 Tenant resource creation

Figure 5.2.2.2-1 shows a scenario where a service consumer requests to create a tenant resource.

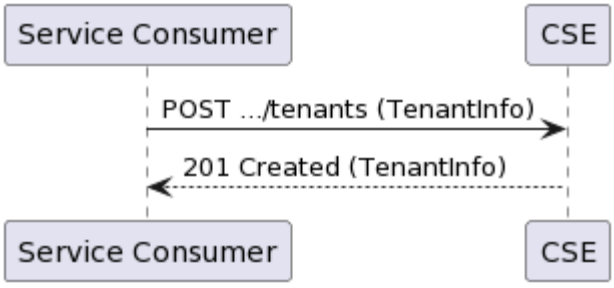

Figure 5.2.2.2-1: Flow of tenant resource creation

Tenant resource creation, as illustrated in Figure 5.2.2.2-1, consists of the following steps:

1) Service consumer sends a request to the CSE to create a tenant resource with the associated information.

2) CSE returns "201 Created" with the message body including the accepted TenantInfo structure.

5.2.2.3 Tenant resource deletion

Figure 5.2.2.3-1 shows a scenario where a service consumer requests to delete a tenant resource.

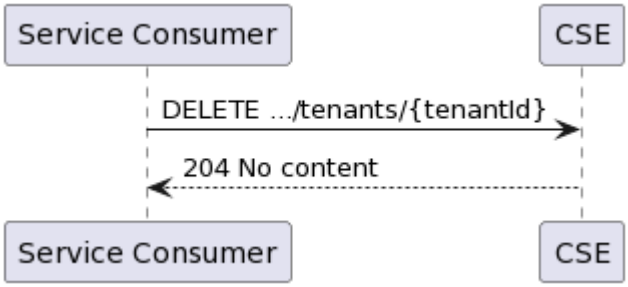

Figure 5.2.2.3-1: Flow of tenant resource deletion

Tenant resource deletion, as illustrated in Figure 5.2.2.3-1, consists of the following steps:

1) Service consumer sends a request to the CSE to delete a tenant resource.

2) CSE responds with "204 No content".

5.2.2.4 Tenant resource update

Figure 5.2.2.4-1 shows a scenario where a service consumer requests to update the information of a tenant resource.

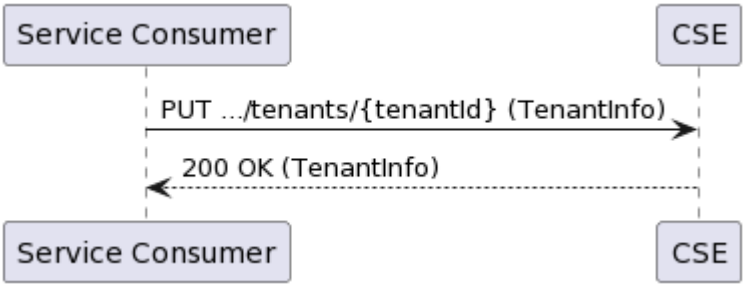

Figure 5.2.2.4-1: Flow of tenant resource update

Tenant resource update, as illustrated in Figure 5.2.2.4-1, consists of the following steps:

1) Service consumer requests the CSE to update the information of a tenant resource by sending a PUT request with the modified data structure specific to that tenant resource.

2) CSE responds with "200 OK" with the message body containing the accepted data structure specific to that tenant resource.

5.2.2.5 Tenant resource query

Figure 5.2.2.5-1 shows a scenario where a service consumer requests to query the information of a specific tenant resource.

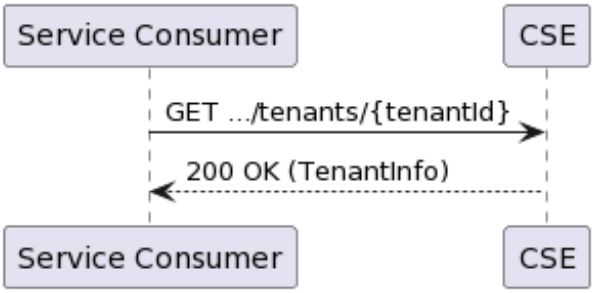

Figure 5.2.2.5-1: Flow of querying a specific tenant resource

Querying a specific tenant resource, as illustrated in Figure 5.2.2.5-1, consists of the following steps:

1) Service consumer requests the CSE to query the information of a specific tenant resource.

2) CSE responds with "200 OK" with the message body containing the information of that tenant resource.

Figure 5.2.2.5-2 shows a scenario where a service consumer requests to query the information of all tenant resources.

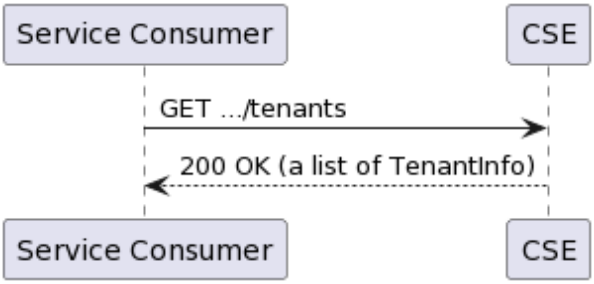

Figure 5.2.2.5-2: Flow of querying all tenant resources

Querying all tenant resources, as illustrated in Figure 5.2.2.5-2, consists of the following steps:

1) Service consumer requests the CSE to query the information of all tenant resources.

2) CSE responds with "200 OK" with the message body containing the information of a list of tenant resources.

5.2.3 Computing resource management

5.2.3.1 Introduction

The computing resource management is about the resource information in an edge site or a MEC system that is associated with a tenant, and the basic functionalities include:

- per tenant resource request in an edge site;
- per tenant resource request update in an edge site;
- per tenant resource quota query in an edge site;
- per tenant resource request in a MEC system;
- per tenant resource request update in a MEC system; and
- per tenant resource quota query in a MEC system.

5.2.3.2 Computing resource request (per-tenant)

Figure 5.2.3.2-1 shows a scenario where a service consumer requests computing resource for a tenant. It is used to request the computing resource in a MEC system or an edge site that can be used for a tenant.

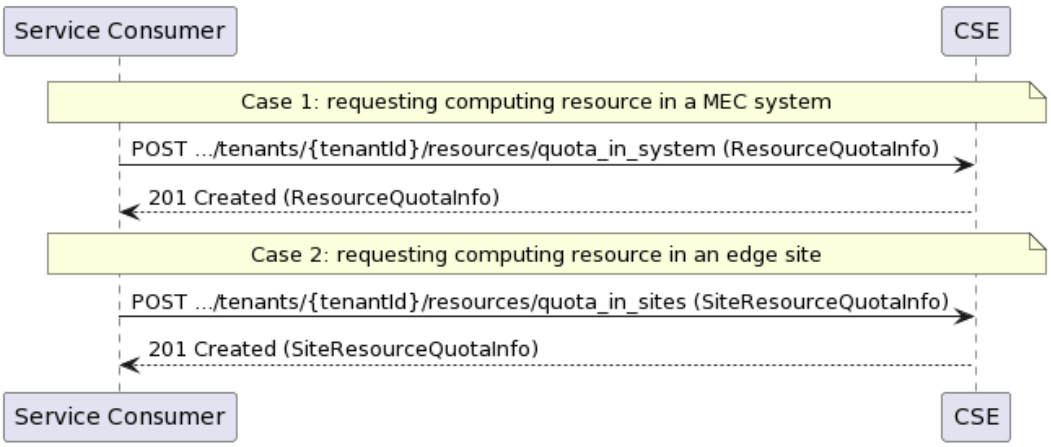

Figure 5.2.3.2-1: Flow of computing resource request

Per-tenant computing resource request, as illustrated in Figure 5.2.3.2-1, consists of the following steps:

1) Service consumer sends a message to the CSE requesting the computing resource quota for a tenant:
   - Case 1: requesting computing resource in a MEC system.
   - Case 2: requesting computing resource in an edge site.
2) CSE returns "201 Created" with the message body including the granted computing resource quota:
   - Case 1: the data structure resourceQuotaInfo is returned.
   - Case 2: the data structure siteResourceQuotaInfo is returned.

5.2.3.3 Computing resource request update (per-tenant)

Figure 5.2.3.3-1 shows a scenario where a service consumer requests to update the computing resource request for a tenant. It is used to update the request of the computing resource in a MEC system or an edge site that can be used for a tenant.

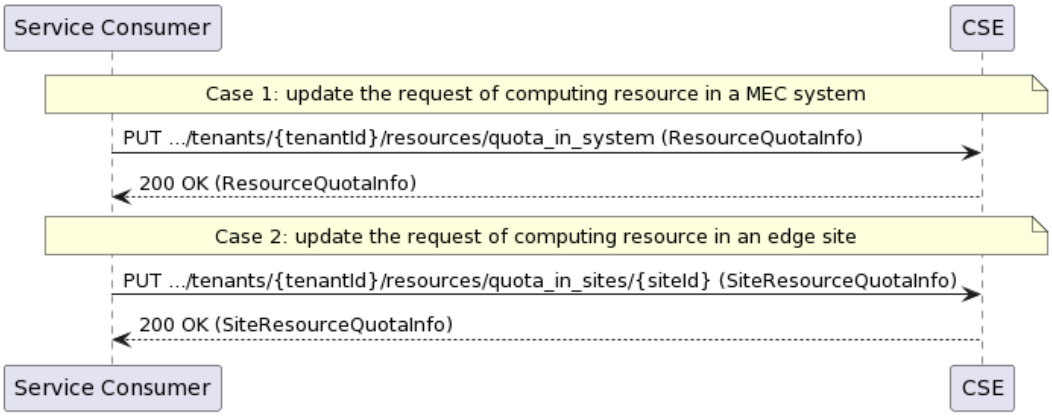

Figure 5.2.3.3-1: Flow of updating the computing resource request

Computing resource request update, as illustrated in Figure 5.2.3.3-1, consists of the following steps:

1) Service consumer requests the CSE to update the computing resource request for a tenant by sending a PUT request with the modified data structure specific to that computing resource request for a tenant:
   - Case 1: updating the request of computing resource in a MEC system.
   - Case 2: updating the request of computing resource in an edge site.
2) CSE responds with "200 OK" with the message body containing the accepted data structure:
   - Case 1: the data structure resourceQuotaInfo is returned.
   - Case 2: the data structure siteResourceQuotaInfo is returned.

5.2.3.4 Computing resource quota query (per tenant)

Figure 5.2.3.4-1 shows a scenario where a service consumer requests to query the computing resource quota for a specific tenant. It is used to query the per-tenant quota of the computing resource in a MEC system or an edge site.

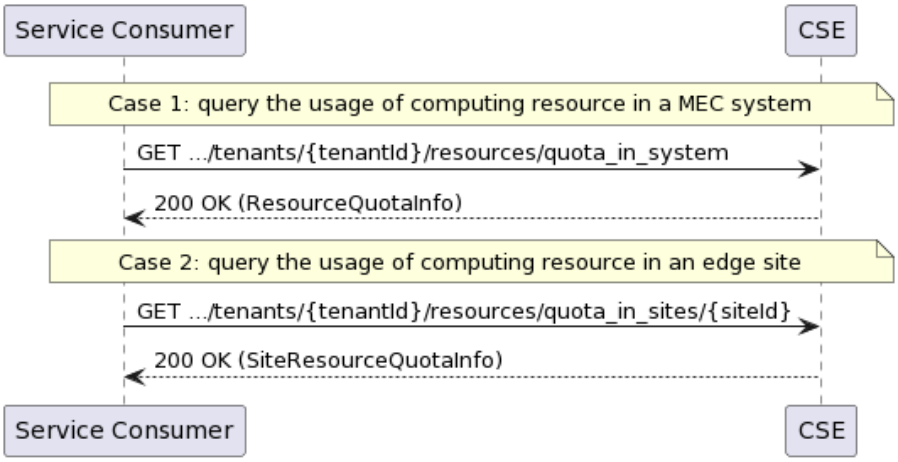

Figure 5.2.3.4-1: Flow of querying the computing resource quota

Querying the per-tenant quota of the computing resource, as illustrated in Figure 5.2.3.4-1, consists of the following steps:

1) Service consumer requests the CSE to query the computing resource quota for a specific tenant:
   - Case 1: query the per-tenant quota of the computing resource in a MEC system.
   - Case 2: query the per-tenant quota of the computing resource in an edge site.
2) CSE responds with "200 OK" with the message body containing the requested information:
   - Case 1: the data structure resourceQuotaInfo is returned.
   - Case 2: the data structure siteResourceQuotaInfo is returned.

5.2.4 REST based subscribe-notify model

5.2.4.1 Subscribing to CSE event notifications

To receive notifications on selected events, the service consumer creates a subscription to certain specific event that is available. Figure 5.2.4.1-1 shows a scenario where the service consumer uses REST based procedures to create a subscription for CSE event notifications.

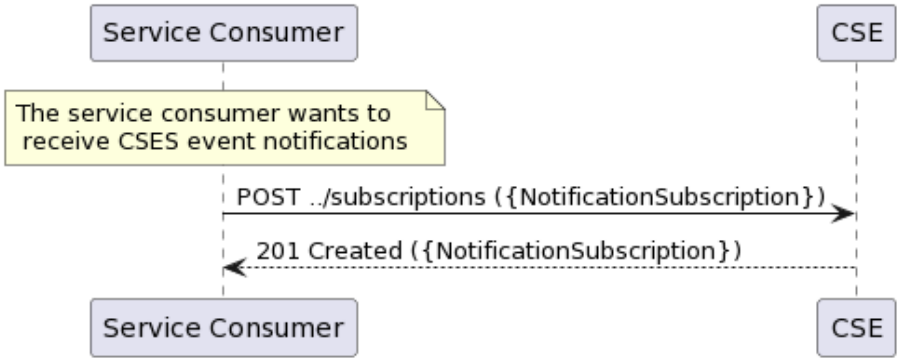

Figure 5.2.4.1-1: Flow of subscribing to CSE event notifications

Subscribing to the CSE event notifications, as illustrated in Figure 5.2.4.1-1, consists of the following steps.

When the service consumer wants to receive CSE event notifications, it creates a subscription:

1) The service consumer sends a POST request with the message body containing the {NotificationSubscription} data structure.

2) CSE sends "201 Created" response with the message body containing the data structure specific to that CSE event subscription.

5.2.4.2 Receiving notification on expiry of CSE event subscription

CSE may define an expiry time for the CSE 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 the expiry, CSE will also send a notification to the service consumer that owns the subscription.

Figure 5.2.4.2-1 shows a scenario where the service consumer receives a subscription expiry notification for the existing subscription.

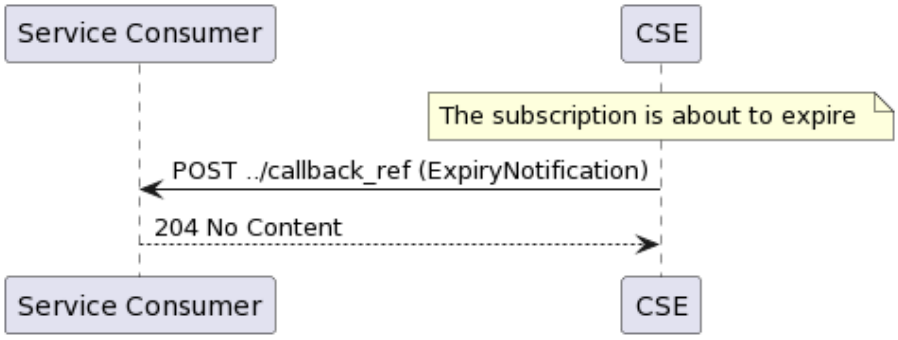

Figure 5.2.4.2-1: Flow of CSE sending a notification on expiry of the subscription

Sending a notification on expiry of the subscription, as illustrated in Figure 5.2.4.2-1 consists of the following steps. If CSE has defined an expiry time for the subscription, CSE will send a notification prior the expiry:

1) CSE 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.

5.2.4.3 Updating subscription for CSE event notifications

Figure 5.2.4.3-1 shows a scenario where the service consumer needs to update an existing subscription for a CSE 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.

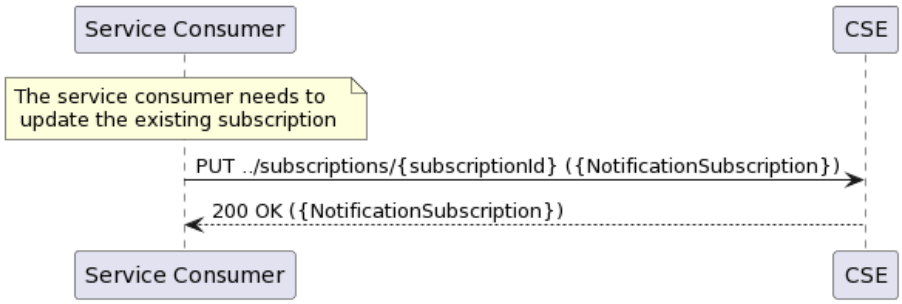

Figure 5.2.4.3-1: Flow of service consumer updating subscription for CSE event notifications

Updating subscription for CSE event notifications, as illustrated in Figure 5.2.4.3-1, consists of the following steps.

When the service consumer needs to modify an existing subscription for CSE 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 containing all the subscription information with the modified data structure specific to that CSE event subscription.

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

5.2.4.4 Unsubscribing from CSE event notifications

When the service consumer does not want to receive notifications anymore after subscribing to CSE events, the service consumer unsubscribes from the CSE event notifications. Figure 5.2.4.4-1 shows a scenario where the service consumer uses REST based procedures to delete the subscription for CSE event notifications.

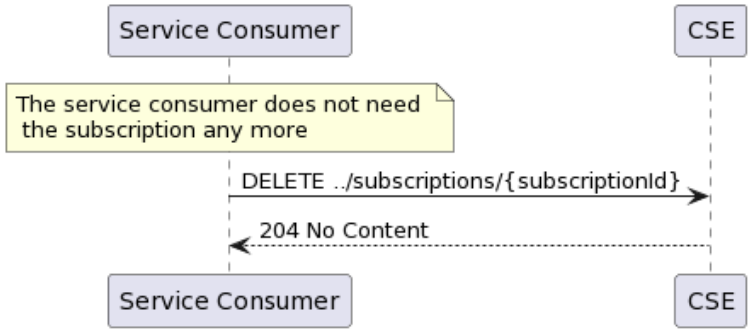

Figure 5.2.4.4-1: Flow of unsubscribing from the CSE event notifications

Unsubscribing from the CSE event notifications, as illustrated in Figure 5.2.4.4-1, consists of the following steps.

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

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

2) CSE sends "204 No content" response.

5.2.5 Receiving CSE event notifications about resource usage in a MEC system

Figure 5.2.5-1 presents the scenario where the CSE sends to the service consumer CSE event notifications about the computing resource usage of a tenant in a MEC system.

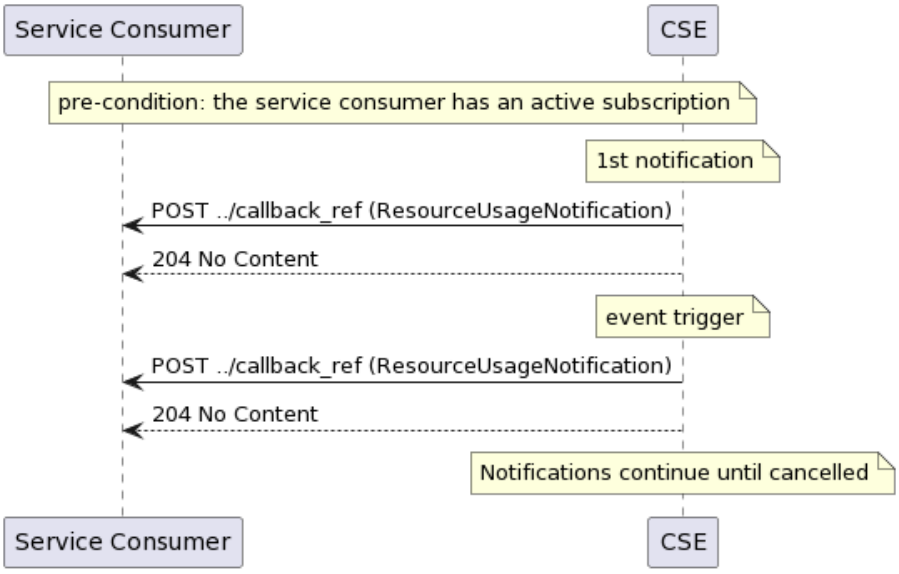

Figure 5.2.5-1: Flow of receiving CSE event notifications on resource usage in a MEC system

Receiving CSE event notifications on resource usage in a MEC system, as illustrated in Figure 5.2.5-1, consists of the following steps:

1) CSE sends a POST request with the message body containing the ResourceUsageNotification data structure to the callback reference address included by the service consumer in the CSE event subscription.

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

5.2.6 Receiving CSE event notifications about resource usage in edge sites

Figure 5.2.6-1 presents the scenario where the CSE sends to the service consumer CSE event notifications about the computing resource usage of a tenant in one or multiple edge sites.

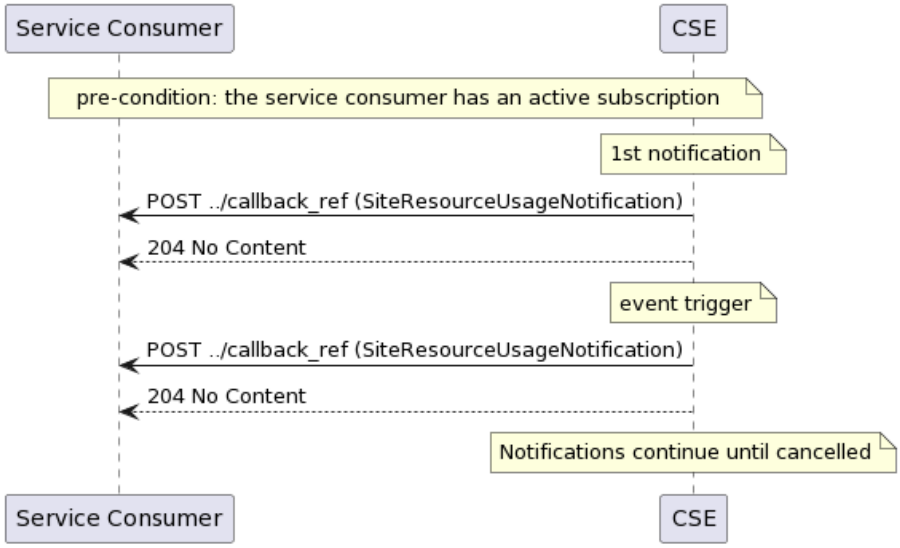

Figure 5.2.6-1: Flow of receiving CSE event notifications on resource usage in edge sites

Receiving CSE event notifications on resource usage in edge sites, as illustrated in Figure 5.2.6-1, consists of the following steps:

1) CSE sends a POST request with the message body containing the SiteResourceUsageNotification data structure to the callback reference address included by the service consumer in the CSE event subscription.

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

6 Data model

6.1 Introduction

The following clauses define the description of the data model.

6.2 Resource data types

6.2.1 Overview

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

6.2.2 Type: TenantInfo

This type represents the information on a tenant.

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

Table 6.2.2-1: Attributes of TenantInfo

Attribute name	Data type	Cardinality	Description
customerId	String	1	Identifier of the customer. For the uniqueness of the identifier across the MEC system, UUID format [i.2] is recommended.
customerName	String	1	Name of the customer. Examples include the name of a company.

Attribute name	Data type	Cardinality	Description
customerCategory	String	0..1	Category of the customer. The example values include: 1. "Finance". 2. "Manufacturing". 3. "Retail". 4. "Education". 5. "Automotive".
tenantId	String	0..1	Identifier of the tenant. Shall be absent in POST requests, and present otherwise.
tenantName	String	1	Name of the tenant.
resourceUseInfo	ResourceInfo	0..1	Resource in the MEC system that can be used by the tenant.
siteList	array(SiteInfo)	0..N	A list of edge sites that can be used by the tenant.
NOTE: Either resourceUseInfo or siteList or none of them shall be present.			

6.2.3 Type: ResourceQuotaInfo

This type represents the information on the computing resource quota for a tenant in a MEC system.

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

Table 6.2.3-1: Attributes of ResourceQuotaInfo

Attribute name	Data type	Cardinality	Description
cpuQuota	Integer	0..1	Allowed number of CPUs in the MEC system that can be used by the tenant.
memoryQuota	Integer	0..1	Allowed amount of memory (MB) in the MEC system that can be used by the tenant.
diskQuota	Integer	0..1	Allowed amount of disk (GB) in the MEC system that can be used by the tenant.
NOTE: At least one of cpuQuota, memoryQuota and, diskQuota shall be present.			

6.2.4 Type: SiteResourceQuotaInfo

This type represents the information on the computing resource quota for a tenant in an edge site.

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

Table 6.2.4-1: Attributes of SiteResourceQuotaInfo

Attribute name	Data type	Cardinality	Description
siteId	String	1	Identifier of an edge site. For the uniqueness of the identifier across the MEC system, UUID format [i.2] is recommended.
cpuQuota	Integer	0..1	Allowed number of CPUs in the edge site that can be used by the tenant.
memoryQuota	Integer	0..1	Allowed amount of memory (MB) in the edge site that can be used by the tenant.
diskQuota	Integer	0..1	Allowed amount of disk (GB) in the edge site that can be used by the tenant.
NOTE: At least one of cpuQuota, memoryQuota and, diskQuota shall be present.			

6.3 Subscription data types

6.3.1 Introduction

This clause defines data structures for subscriptions.

6.3.2 Type: ResourceUsageSubscription

This type represents a subscription to get updates on the computing resource usage of a tenant in a MEC system.

Table 6.3.2-1: Attributes of the ResourceUsageSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "ResourceUsageSubscription".
callbackReference	Uri	0..1	URI exposed by the client on which to receive notifications via HTTP. See note 1.
requestTestNotification	Boolean	0..1	Set to TRUE by the service consumer to request a test notification via HTTP on the callbackReference URI, as specified in ETSI GS MEC 009 [1], clause 6.12a.
websockNotifConfig	WebsockNotifConfig	0..1	Provides details to negotiate and signal the use of a Websocket connection between the WAIS and the service consumer for notifications. See note 1.
_links	Structure (inline)	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 CSE API as it acts as an ID for the subscription (SubscriptionId).
customerId	String	1	Identifier to uniquely specify the customer for the subscription.
tenantId	String	1	Identifier of the tenant.
notificationTrigger	Structure (inline)	1	Set for trigger-based event notification reporting.
>triggerType	Enum (inline)	1	Trigger for the notification: 10 = Notification trigged based on the number of used CPUs. 11 = Notification trigged based on the number of remaining CPUs. 20 = Notification trigged based on the amount of used memory. 21 = Notification trigged based on the amount of remaining memory. 30 = Notification trigged based on the amount of used disk. 31 = Notification trigged based on the amount of remaining disk.
>threshold	Uint8	1	Threshold for trigger-based event reporting.
>greaterOrLess	Boolean	1	Indicator for the triggering condition: 0: greater than or equal to the threshold. 1: less than or equal to the threshold.
expiryDeadline	TimeStamp	0..1	The expiration time of the subscription determined by the CSE.
NOTE: At least one of callbackReference and websockNotifConfig shall be provided by the service consumer. If both are provided, it is up to CSE to select an alternative and return only that alternative in the response, as specified in ETSI GS MEC 009 [1], clause 6.12a.			

6.3.3 Type: SiteResourceUsageSubscription

This type represents a subscription to get updates on the computing resource usage of a tenant in one or multiple edge sites.

Table 6.3.3-1: Attributes of the SiteResourceUsageSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "SiteResourceUsageSubscription".
callbackReference	Uri	0..1	URI exposed by the client on which to receive notifications via HTTP. See note 1.
requestTestNotification	Boolean	0..1	Set to TRUE by the service consumer to request a test notification via HTTP on the callbackReference URI, as specified in ETSI GS MEC 009 [1], clause 6.12a.
websockNotifConfig	WebsockNotifConfig	0..1	Provides details to negotiate and signal the use of a Websocket connection between the WAIS 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 CSE API as it acts as an ID for the subscription (SubscriptionId).
customerId	String	1	Identifier to uniquely specify the customer for the subscription.
tenantId	String	1	Identifier of the tenant.
siteList	Array(siteId)	1	Identifiers to uniquely specify a list of edge sites for the subscription.
notificationTrigger	Structure (inline)	1	Set for trigger-based event notification reporting.
>triggerType	Enum (inline)	1	Trigger for the notification: 10 = Notification trigged based on the number of used CPUs. 11 = Notification trigged based on the number of remaining CPUs. 20 = Notification trigged based on the amount of used memory. 21 = Notification trigged based on the amount of remaining memory. 30 = Notification trigged based on the amount of used disk. 31 = Notification trigged based on the amount of remaining disk.
>threshold	Uint8	1	Threshold for trigger-based event reporting.
>greaterOrLess	Boolean	1	Indicator for the triggering condition: 0: greater than or equal to the threshold. 1: less than or equal to the threshold.
expiryDeadline	TimeStamp	0..1	The expiration time of the subscription determined by the CSE.
NOTE: At least one of callbackReference and websockNotifConfig shall be provided by the service consumer. If both are provided, it is up to CSE to select an alternative and return only that alternative in the response, as specified in ETSI GS MEC 009 [1], clause 6.12a.			

6.3.4 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 6.3.4-1: Attributes of the SubscriptionLinkList

Attribute name	Data type	Cardinality	Description
_links	Structure (inlined)	1	List of hyperlinks related to the resource.
>self	LinkType	1	Self-referring URI.
subscription	Structure (inlined)	0..N	
>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: "ResourceUsageSubscription" "SiteResourceUsageSubscription"

6.4 Notifications data types

6.4.1 Introduction

This clause defines data structures that define notifications.

6.4.2 Type: ResourceUsageNotification

This type represents a notification from CSE with regards to the computing resource usage of a tenant in a MEC system.

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

Table 6.4.2-1: Attributes of the ResourceUsageNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "ResourceUsageNotification".
timeStamp	TimeStamp	0..1	Time stamp.
customerId	String	1	Identifier to uniquely specify the customer for the subscription.
tenantId	String	1	Identifier of the tenant.
resourceUseInfo	Structure (inlined)	1	Resource usage in the MEC system by the tenant.
>cpuUsed	Integer	0..1	Used number of CPUs by the tenant.
>cpuRemain	Integer	0..1	Remaining number of CPUs that can be used by the tenant.
>memoryUsed	Integer	0..1	Used amount of memory (MB) by the tenant.
>memoryRemain	Integer	0..1	Remaining amount of memory (MB) that can be used by the tenant.
>diskUsed	Integer	0..1	Used amount of disk (GB) by the tenant.
>diskRemain	Integer	0..1	Remaining amount of disk (GB) that can be used by the tenant.
NOTE: The corresponding resource usage information shall be included based on the subscription.			

6.4.3 Type: SiteResourceUsageNotification

This type represents a notification from CSE with regards to the computing resource usage of a tenant in one or multiple edge sites.

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

Table 6.4.3-1: Attributes of the SiteResourceUsageNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "SiteResourceUsageNotification".
timeStamp	TimeStamp	0..1	Time stamp.
customerId	String	1	Identifier to uniquely specify the customer for the subscription.
tenantId	String	1	Identifier of the tenant.
resourceUseInfo	Structure (inlined)	1..N	Resource usage in the edge sites by the tenant.
>siteId	String	1	Identifier of an edge site.
>>cpuUsed	Integer	0..1	Used number of CPUs by the tenant.
>>cpuRemain	Integer	0..1	Remaining number of CPUs that can be used by the tenant.
>>memoryUsed	Integer	0..1	Used amount of memory (MB) by the tenant.
>>memoryRemain	Integer	0..1	Remaining amount of memory (MB) that can be used by the tenant.
>>diskUsed	Integer	0..1	Used amount of disk (GB) by the tenant.
>>diskRemain	Integer	0..1	Remaining amount of disk (GB) that can be used by the tenant.
NOTE: The corresponding resource usage information shall be included based on the subscription.			

6.4.4 Type: ExpiryNotification

This type represents a notification from CSE with regards to expiry of an existing subscription.

The Notification is sent by the CSE about expiry of a subscription.

Table 6.4.4-1: Attributes of the ExpiryNotification

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

6.5 Referenced structured data types

6.5.1 Introduction

This clause defines data structures that can be referenced from data structures defined in the previous clauses, but can neither be resource representations nor notifications.

6.5.2 Type: SiteInfo

This type represents the information on an edge site.

The attributes of the SiteInfo shall follow the notations provided in Table 6.5.2-1.

Table 6.5.2-1: Attributes of SiteInfo

Attribute name	Data type	Cardinality	Description
siteId	String	1	Identifier of an edge site. For the uniqueness of the identifier across the MEC system, UUID format [i.2] is recommended.
resourceInfo	ResourceInfo	0..1	Resource information in the edge site that is associated with a specific tenant.

6.5.3 Type: ResourceInfo

This type represents the resource information in an edge site or a MEC system that is associated with a tenant.

The attributes of the ResourceInfo shall follow the notations provided in Table 6.5.3-1.

Table 6.5.3-1: Attributes of ResourceInfo

Attribute name	Data type	Cardinality	Description
cpuQuota	Integer	0..1	Allowed number of CPUs that can be used by the tenant.
cpuUsed	Integer	0..1	Used number of CPUs by the tenant.
cpuRemain	Integer	0..1	Remaining number of CPUs that can be used by the tenant.
memoryQuota	Integer	0..1	Allowed amount of memory (MB) that can be used by the tenant.
memoryUsed	Integer	0..1	Used amount of memory (MB) by the tenant.
memoryRemain	Integer	0..1	Remaining amount of memory (MB) that can be used by the tenant.
diskQuota	Integer	0..1	Allowed amount of disk (GB) that can be used by the tenant.
diskUsed	Integer	0..1	Used amount of disk (GB) by the tenant.
diskRemain	Integer	0..1	Remaining amount of disk (GB) that can be used by the tenant.

7 API definitions

7.1 Introduction

This clause defines the resources and operations of the Customer Self-service Enablement (CSE) API.

7.2 Global definitions and resource structure

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

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

"ApiRoot" and "apiName" are discovered using the service registry. It includes the scheme ("https"), host and optional port, and an optional prefix string.

The API shall support HTTP over TLS as defined in clause 6.22 of ETSI GS MEC 009 [1].

The "apiVersion" shall be set to "v1" for the current version of the present document. All resource URIs in the clauses below are defined relative to the above root URI.

The content format JSON shall be supported.

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

This API shall use OAuth 2.0, as defined in clause 6.16 of ETSI GS MEC 009 [1]. This OAuth 2.0 authorization procedure shall occur only on TLS-protected connections.

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 [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.

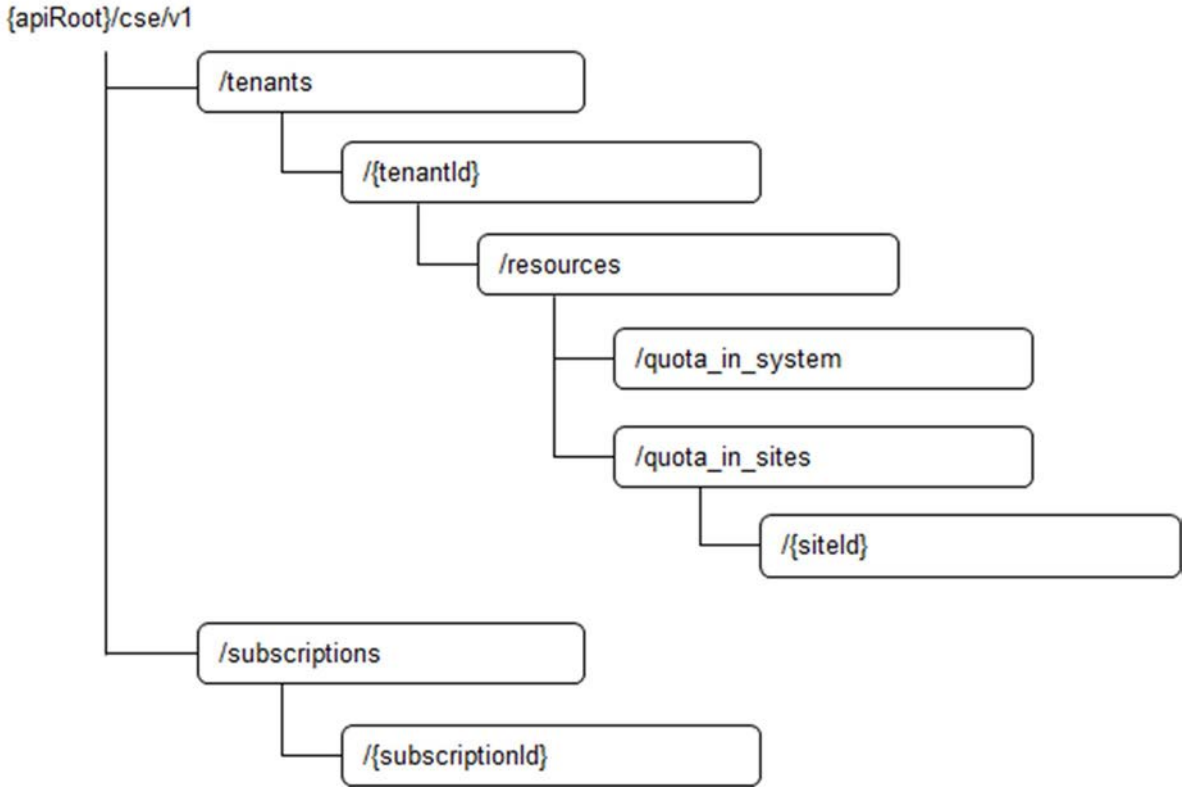

Figure 7.2-1: Resource URI structure of the CSE API

Table 7.2-1: Resources and methods overview

Resource name	Resource URI	HTTP method	Meaning
A list of tenants	/tenants	GET	Retrieve information about a list of tenants
		POST	Create a tenant resource representation
Individual tenant	/tenants/{tenantId}	GET	Retrieve information about a specific tenant resource representation
		PUT	Modify the information about a specific tenant resource representation
		DELETE	Delete a specific tenant resource representation
Per system resource quota	/tenants/{tenantId}/resources/quota_in_system	POST	Create a resourceQuotaInfo resource
		PUT	Modify the information about a resourceQuotaInfo resource
		GET	Retrieve information about a resourceQuotaInfo resource
A list of per site resource quota	/tenants/{tenantId}/resources/quota_in_sites	POST	Create a siteResourceQuotaInfo resource
		GET	Retrieve information about a list of siteResourceQuotaInfo resources
Individual per site resource quota	/tenants/{tenantId}/resources/quota_in_sites/{siteId}	GET	Retrieve information about a siteResourceQuotaInfo resource
		PUT	Modify the information about a siteResourceQuotaInfo resource
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 the existing subscription of a tenant
		PUT	Modify existing subscription by sending a new data structure
		DELETE	Cancel an existing subscription
Notification callback	Client provided callback reference	POST	Send a notification

7.3 Resource: a list of tenants

7.3.1 Description

This resource is used to represent a list of tenants.

7.3.2 Resource definition

Resource URI:

{apiRoot}/cse/v1/tenants

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 "a list of tenants"

Name	Definition
apiRoot	See clause 7.2.

7.3.3 Resource methods

7.3.3.1 GET

The GET method is used to retrieve information about a list of tenantInfo resource representations.

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

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

Name	Data type	Cardinality	Remarks
customerId	String	1	Identifier of the customer.
customerName	String	1	Name of the customer.
tenantId	String	0..N	Multiple tenantId may be used as an input parameter to query the availability of a list of tenants.
tenantName	String	0..N	Multiple tenantName may be used as an input parameter to query the availability of a list of tenants.

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**
	TenantInfo	0..N	200 OK	Upon success, a response message content containing an array of the tenantInfo is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	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	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	414 URI Too Long	It is used to indicate that the server is refusing to process the request because the request URI is longer than the server is willing or able to process.

7.3.3.2 PUT

Not supported.

7.3.3.3 PATCH

Not supported.

7.3.3.4 POST

The POST method is used to create a tenant resource representation.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in Tables 7.3.3.4-1 and 7.3.3.4-2.

Table 7.3.3.4-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.3.4-2: Data structures supported by the POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	TenantInfo	1	Message content in the request contains TenantInfo to be created.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	TenantInfo	1	201 Created	Upon success, the HTTP response shall include a "Location" HTTP header that contains the resource URI of the created resource.
	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	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	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.

7.3.3.5 DELETE

Not supported.

7.4 Resource: individual tenant

7.4.1 Description

This resource is used to represent a tenant.

7.4.2 Resource definition

Resource URI:

{apiRoot}/cse/v1/tenants/{tenantId}

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 "individual tenant"

Name	Definition
apiRoot	See clause 7.2.
tenantId	Tenant identifier.

7.4.3 Resource methods

7.4.3.1 GET

The GET method is used to retrieve information about a tenant resource representation.

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

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

Name	Data type	Cardinality	Remarks
n/a			

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**
	TenantInfo	1	200 OK	Upon success, a response message content containing a representation of the resource is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	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	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.

7.4.3.2 PUT

The PUT method is used to modify the information of a tenant resource representation.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in Tables 7.4.3.2-1 and 7.4.3.2-2.

Table 7.4.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.3.2-2: Data structures supported by the PUT request/response on this resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	TenantInfo	1	Message content in the request contains TenantInfo with the updated information.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	TenantInfo	1	200 OK	Upon success, the HTTP response shall include a "Location" HTTP header that contains the resource URI of the updated resource.
	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	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	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	412 Precondition Failed	It is used when a condition has failed during conditional requests, e.g. when using ETags to avoid write conflicts. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.3.3 PATCH

Not supported.

7.4.3.4 POST

Not supported.

7.4.3.5 DELETE

The DELETE method is used to delete a tenant resource representation.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in Tables 7.4.3.5-1 and 7.4.3.5-2.

Table 7.4.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.3.5-2: Data structures supported by the DELETE request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a	1	204 No Content	The operation has been successful. The response message content shall be empty.
	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	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.

7.5 Resource: per system resource quota

7.5.1 Description

This resource is used to represent a per system resource quota.

7.5.2 Resource definition

Resource URI:

{apiRoot}/cse/v1/tenants/{tenantId}/resources/quota_in_system

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 "per system resource quota"

Name	Definition
apiRoot	See clause 7.2.
tenantId	Tenant identifier.

7.5.3 Resource methods

7.5.3.1 GET

The GET method is used to retrieve information about a resourceQuotaInfo resource.

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
n/a			

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

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	ResourceQuotaInfo	1	200 OK	Upon success, a response message content containing the resourceQuotaInfo is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	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	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	414 URI Too Long	It is used to indicate that the server is refusing to process the request because the request URI is longer than the server is willing or able to process.

7.5.3.2 PUT

The PUT method is used to modify the information of a resourceQuotaInfo resource.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in Tables 7.5.3.2-1 and 7.5.3.2-2.

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

Name	Data type	Cardinality	Remarks
n/a			

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

Request body	**Data type**	**Cardinality**	**Remarks**	
	ResourceQuotaInfo	1	Message content in the request contains resourceQuotaInfo with the updated information.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	ResourceQuotaInfo	1	200 OK	Upon success, the HTTP response shall include a "Location" HTTP header that contains the resource URI of the updated resource.
	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	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	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	412 Precondition Failed	It is used when a condition has failed during conditional requests, e.g. when using ETags to avoid write conflicts. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.5.3.3 PATCH

Not supported.

7.5.3.4 POST

The POST method is used to create a resourceQuotaInfo resource.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in Tables 7.5.3.4-1 and 7.5.3.4-2.

Table 7.5.3.4-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.5.3.4-2: Data structures supported by the POST request/response on this resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	ResourceQuotaInfo	1	Message content in the request contains resourceQuotaInfo to be created.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	ResourceQuotaInfo	1	201 Created	Upon success, the HTTP response shall include a "Location" HTTP header that contains the resource URI of the created resource.
	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	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	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.

7.5.3.5 DELETE

Not supported.

7.6 Resource: a list of per site resource quota

7.6.1 Description

This resource is used to represent a list of per site resource quota.

7.6.2 Resource definition

Resource URI:

{apiRoot}/cse/v1/tenants/{tenantId}/resources/quota_in_sites

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 "a list of per site resource quota"

Name	Definition
apiRoot	See clause 7.2.
tenantId	Tenant identifier.

7.6.3 Resource methods

7.6.3.1 GET

The GET method is used to retrieve information about a list of siteResourceQuotaInfo resources.

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
siteId	String	0..N	Multiple siteId may be used as an input parameter to query the availability of a list of siteResourceQuotaInfo resources.

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**
	SiteResourceQuotaInfo	0..N	200 OK	Upon success, a response message content containing an array of the siteResourceQuotaInfo resources is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	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	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.

7.6.3.2 PUT

Not supported.

7.6.3.3 PATCH

Not supported.

7.6.3.4 POST

The POST method is used to create a siteResourceQuotaInfo resource.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in Tables 7.6.3.4-1 and 7.6.3.4-2.

Table 7.6.3.4-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.6.3.4-2: Data structures supported by the POST request/response on this resource

	Data type	Cardinality	Remarks	
Request body	SiteResourceQuotaInfo	1	Message content in the request contains siteResourceQuotaInfo to be created.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SiteResourceQuotaInfo	1	201 Created	Upon success, the HTTP response shall include a "Location" HTTP header that contains the resource URI of the created resource.
	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	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	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.

7.6.3.5 DELETE

Not supported.

7.7 Resource: individual per site resource quota

7.7.1 Description

This resource is used to represent a per site resource quota.

7.7.2 Resource definition

Resource URI:

{apiRoot}/cse/v1/tenants/{tenantId}/resources/quota_in_sites/{siteId}

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 "individual per site resource quota"

Name	Definition
apiRoot	See clause 7.2.
tenantId	Tenant identifier.
siteId	Site identifier.

7.7.3 Resource methods

7.7.3.1 GET

The GET method is used to retrieve information about a siteResourceQuotaInfo resource.

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
n/a			

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

	Data type	Cardinality	Response Codes	Remarks
Request body	n/a			
Response body	SiteResourceQuotaInfo	1	200 OK	Upon success, a response message content containing a siteResourceQuotaInfo resource is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	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	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.

7.7.3.2 PUT

The PUT method is used to modify the information of a siteResourceQuotaInfo resource.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in Tables 7.7.3.2-1 and 7.7.3.2-2.

Table 7.7.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.7.3.2-2: Data structures supported by the PUT request/response on this resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	SiteResourceQuotaInfo	1	Message content in the request contains siteResourceQuotaInfo with the updated information.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	SiteResourceQuotaInfo	1	200 OK	Upon success, the HTTP response shall include a "Location" HTTP header that contains the resource URI of the updated resource.
	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	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	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	412 Precondition Failed	It is used when a condition has failed during conditional requests, e.g. when using ETags to avoid write conflicts. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.7.3.3 PATCH

Not supported.

7.7.3.4 POST

Not supported.

7.7.3.5 DELETE

Not supported.

7.8 Resource: subscriptions

7.8.1 Description

This resource contains various resources related to subscriptions for notifications.

7.8.2 Resource definition

Resource URI:

{apiRoot}/cse/v1/subscriptions/

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 "subscriptions"

Name	Definition
apiRoot	See clause 7.2

7.8.3 Resource methods

7.8.3.1 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 7.8.3.1-1 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
subscription_type	String	0..1	Query parameter to filter on a specific subscription type. Permitted values: ResourceUsageSubscription SiteResourceUsageSubscription
tenantId	String	0..1	Query parameter to filter on a specific tenant.

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**
	SubscriptionLinkList	1	200 OK	Upon success, a response body containing the list of links to requestor's subscriptions is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide 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

Not supported.

7.8.3.3 PATCH

Not supported.

7.8.3.4 POST

The POST method is used to create a new subscription to CSE 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 7.8.3.4-1.

Table 7.8.3.4-1: Data structures supported by the POST request/response on this resource

	Data type	Cardinality	Response Codes	Remarks
Request body	{NotificationSubscription}	1		The entity body in the request contains data type of the specific CSE event subscription that is to be created, where the data type options are listed below: ResourceUsageSubscription SiteResourceUsageSubscription
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	{NotificationSubscription}	1	201 Created	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: ResourceUsageSubscription SiteResourceUsageSubscription The created resource is identified by its subscriptionId included in the "self" attribute (self-referring URI).
	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	415 Unsupported Media Type	It is used to indicate that the server or the client does not support the content type of the entity body. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	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.

7.8.3.5 DELETE

Not supported.

7.9 Resource: existing subscription

7.9.1 Description

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

7.9.2 Resource definition

Resource URI:

{apiRoot}/cse/v1/subscriptions/{subscriptionId}

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

Table 7.9.2-1: Resource URI variables for resource "existing subscription"

Name	Definition
apiRoot	See clause 7.2.
subscriptionId	Refers to a created subscription, where the CSE allocates a unique resource name for this subscription and indicates the resource name in the "self" attribute of the subscription type.

7.9.3 Resource methods

7.9.3.1 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 7.9.3.1-1 and 7.9.3.1-2.

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

Name	Data type	Cardinality	Remarks
n/a			

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

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	{NotificationSubscription}	1	200 OK	Upon success, a response body containing data type describing the specific CSE event subscription is returned. The allowed data types for subscriptions are: ResourceUsageSubscription SiteResourceUsageSubscription
	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.

7.9.3.2 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 7.9.3.2-1 and 7.9.3.2-2.

Table 7.9.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.9.3.2-2: Data structures supported by the PUT request/response on this resource

	Data type	Cardinality	Response Codes	Remarks
Request body	{NotificationSubscription}	1		New NotificationSubscription is included as entity body of the request. The allowed data types for subscriptions are: ResourceUsageSubscription SiteResourceUsageSubscription
Response body	{NotificationSubscription}	1	200 OK	Upon success, a response body containing data type describing the updated subscription is returned. The allowed data types for subscriptions are: ResourceUsageSubscription SiteResourceUsageSubscription
	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	412 Precondition Failed	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.
	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.

7.9.3.3 PATCH

Not supported.

7.9.3.4 POST

Not supported.

7.9.3.5 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 7.9.3.5-1 and 7.9.3.5-2.

Table 7.9.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.9.3.5-2: Data structures supported by the DELETE request/response on this resource

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

8 OpenAPI definitions (informative)

8.1 Introduction

To complement the definitions for each method and resource defined in the above clauses of the present document, this clause provides supplementary codes to illustrate the usage of the defined data model.

8.2 Definition and operation on Resources

Example:

```
openapi: 3.1.0
info:
  title: ETSI GS MEC 048 Tenant Management API
  description: |-
    Tenant management API defined in ETSI GS MEC 048.
  license:
    name: BSD-3-Clause
    url: https://forge.etsi.org/legal-matters
  version: 3.1.1
  contact:
    name: ETSI Forge
    url: https://forge.etsi.org/rep/mec/gs048-cse-api
    email: cti_support@etsi.org
servers:
  - url: https://localhost/cse/v1
tags:
  - name: tenantMgt
    description: tenant management.
  - name: tenantSubscription
    description: subscription and notification related to tenant management.
  - name: quotaInSystem
    description: per system resource quota.
  - name: quotaInSite
    description: per site resource quota.
paths:
```

```
/tenants:
  get:
    tags:
      - tenantMgt
    summary: retrieve information about a list of tenants
    description: The GET method is used to retrieve information about a list of tenantInfo
    resource representations.
    operationId: tenantsGET
    parameters:
      - $ref: '#/components/parameters/CustomerId'
      - $ref: '#/components/parameters/CustomerName'
      - $ref: '#/components/parameters/QueryTenantId'
      - $ref: '#/components/parameters/TenantName'
    responses:
      '200':
        description: Successful query of the tenants information.
        content:
          application/json:
            schema:
              type: array
              items:
                $ref: '#/components/schemas/TenantInfo'
      '400':
        $ref: '#/components/responses/400'
      '404':
        $ref: '#/components/responses/404'
      '403':
        $ref: '#/components/responses/403'
      '414':
        $ref: '#/components/responses/414'
  post:
    tags:
      - tenantMgt
    summary: create a tenant resource representation
    description: The POST method is used to create a tenant resource representation.
    operationId: tenantsPOST
    requestBody:
      description: Message content in the request contains TenantInfo to be created.
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/TenantInfo'
      required: true
    responses:
      '201':
        description: Successful creation of the tenant.
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/TenantInfo'
      '400':
        $ref: '#/components/responses/400'
      '404':
        $ref: '#/components/responses/404'
      '403':
        $ref: '#/components/responses/403'
/tenants/{tenantId}:
  parameters:
    - $ref: '#/components/parameters/TenantId'
  put:
    tags:
      - tenantMgt
    summary: Modify the information about a specific tenant resource representation
    description: The PUT method is used to modify the information of a tenant resource
    representation.
    operationId: tenantPUT
    requestBody:
      description: the TenantInfo with the updated information.
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/TenantInfo'
      required: true
    responses:
      '200':
        description: Successful operation.
        content:
          application/json:
```

```
            schema:
              $ref: '#/components/schemas/TenantInfo'
      '404':
        $ref: '#/components/responses/404'
      '400':
        $ref: '#/components/responses/400'
      '403':
        $ref: '#/components/responses/403'
      '412':
        $ref: '#/components/responses/412'
  get:
    tags:
      - tenantMgt
    summary: Retrieve information about a specific tenant resource representation
    description: The GET method is used to retrieve information about a tenant resource
    representation.
    operationId: tenantGET
    responses:
      '200':
        description: successful operation.
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/TenantInfo'
      '400':
        $ref: '#/components/responses/400'
      '404':
        $ref: '#/components/responses/404'
      '403':
        $ref: '#/components/responses/403'
  delete:
    tags:
      - tenantMgt
    summary: Delete a specific tenant resource representation
    description: The DELETE method is used to delete a tenant resource representation.
    operationId: tenantDELETE
    responses:
      '204':
        description: successful operation.
      '404':
        $ref: '#/components/responses/404'
      '403':
        $ref: '#/components/responses/403'
/tenants/{tenantId}/resources/quota_in_system:
  parameters:
    - $ref: '#/components/parameters/TenantId'
  post:
    tags:
      - quotaInSystem
    summary: Create a resourceQuotaInfo resource
    description: The POST method is used to create a resourceQuotaInfo resource.
    operationId: quotaSystemPOST
    requestBody:
      description: the resourceQuotaInfo to be created.
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/ResourceQuotaInfo'
      required: true
    responses:
      '201':
        description: Successful operation.
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ResourceQuotaInfo'
      '400':
        $ref: '#/components/responses/400'
      '404':
        $ref: '#/components/responses/404'
      '403':
        $ref: '#/components/responses/403'
  put:
    tags:
      - quotaInSystem
    summary: Modify the information about a resourceQuotaInfo resource
    description: The PUT method is used to modify a resourceQuotaInfo resource.
    operationId: quotaSystemPUT
```

```
    requestBody:
      description: the resourceQuotaInfo with the updated info.
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/ResourceQuotaInfo'
      required: true
    responses:
      '200':
        description: Successful operation.
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ResourceQuotaInfo'
      '400':
        $ref: '#/components/responses/400'
      '404':
        $ref: '#/components/responses/404'
      '403':
        $ref: '#/components/responses/403'
      '412':
        $ref: '#/components/responses/412'
  get:
    tags:
      - quotaInSystem
    summary: Retrieve information about a resourceQuotaInfo resource
    description: The GET method is used to retrieve information about a resourceQuotaInfo
    resource.
    operationId: quotaSystemGET
    responses:
      '200':
        description: successful operation.
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ResourceQuotaInfo'
      '400':
        $ref: '#/components/responses/400'
      '404':
        $ref: '#/components/responses/404'
      '403':
        $ref: '#/components/responses/403'
      '414':
        $ref: '#/components/responses/414'
/tenants/{tenantId}/resources/quota_in_sites:
  parameters:
    - $ref: '#/components/parameters/TenantId'
  get:
    tags:
      - quotaInSite
    summary: Retrieve information about a list of siteResourceQuotaInfo resources
    description: The GET method is used to retrieve information about a list of
    siteResourceQuotaInfo resources.
    operationId: quotaSitesGET
    parameters:
      - $ref: '#/components/parameters/QuerySiteId'
    responses:
      '200':
        description: Successful query of information.
        content:
          application/json:
            schema:
              type: array
              items:
                $ref: '#/components/schemas/SiteResourceQuotaInfo'
      '400':
        $ref: '#/components/responses/400'
      '404':
        $ref: '#/components/responses/404'
      '403':
        $ref: '#/components/responses/403'
  post:
    tags:
      - quotaInSite
    summary: Create a siteResourceQuotaInfo resource
    description: The POST method is used to create a siteResourceQuotaInfo resource.
    operationId: quotaSitesPOST
    requestBody:
```

```
      description: Message content in the request contains siteResourceQuotaInfo to be created.
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SiteResourceQuotaInfo'
      required: true
    responses:
      '201':
        description: Successful creation of the SiteResourceQuotaInfo object.
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/SiteResourceQuotaInfo'
      '400':
        $ref: '#/components/responses/400'
      '404':
        $ref: '#/components/responses/404'
      '403':
        $ref: '#/components/responses/403'
/tenants/{tenantId}/resources/quota_in_sites/{siteId}:
  parameters:
    - $ref: '#/components/parameters/TenantId'
    - $ref: '#/components/parameters/SiteId'
  put:
    tags:
      - quotaInSite
    summary: Modify the information about a siteResourceQuotaInfo resource
    description: The PUT method is used to modify the information of a siteResourceQuotaInfo
    resource.
    operationId: quotaSitePUT
    requestBody:
      description: the SiteResourceQuotaInfo with the updated information.
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SiteResourceQuotaInfo'
      required: true
    responses:
      '200':
        description: Successful operation.
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/SiteResourceQuotaInfo'
      '400':
        $ref: '#/components/responses/400'
      '404':
        $ref: '#/components/responses/404'
      '403':
        $ref: '#/components/responses/403'
      '412':
        $ref: '#/components/responses/412'
  get:
    tags:
      - quotaInSite
    summary: Retrieve information about a siteResourceQuotaInfo resource
    description: The GET method is used to retrieve information about a siteResourceQuotaInfo
    resource.
    operationId: quotaSiteGET
    responses:
      '200':
        description: successful operation.
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/SiteResourceQuotaInfo'
      '400':
        $ref: '#/components/responses/400'
      '404':
        $ref: '#/components/responses/404'
      '403':
        $ref: '#/components/responses/403'
/subscriptions:
  get:
    tags:
      - tenantSubscription
    summary: Retrieve a list of active subscriptions for this subscriber
```

```
    description: The GET method is used to request information about the subscriptions for this
    requestor.
    operationId: subscriptionsGET
    parameters:
      - $ref: '#/components/parameters/SubscriptionType'
      - $ref: '#/components/parameters/TenantId'
    responses:
      '200':
        description: Successful query of the requestor's subscriptions.
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/SubscriptionLinkList'
      '400':
        $ref: '#/components/responses/400'
      '401':
        $ref: '#/components/responses/401'
      '404':
        $ref: '#/components/responses/404'
      '403':
        $ref: '#/components/responses/403'
      '406':
        $ref: '#/components/responses/406'
      '429':
        $ref: '#/components/responses/429'
  post:
    tags:
      - tenantSubscription
    summary: Create a new subscription
    description: The POST method is used to create a new subscription to CSE notifications.
    operationId: subscriptionsPOST
    requestBody:
      description: the specific CSE event subscription that is to be created.
      content:
        application/json:
          schema:
            oneOf:
              - $ref: '#/components/schemas/ResourceUsageSubscription'
              - $ref: '#/components/schemas/SiteResourceUsageSubscription'
      required: true
    responses:
      '201':
        description: Successful creation of the tenant.
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/ResourceUsageSubscription'
                - $ref: '#/components/schemas/SiteResourceUsageSubscription'
      '400':
        $ref: '#/components/responses/400'
      '401':
        $ref: '#/components/responses/401'
      '403':
        $ref: '#/components/responses/403'
      '404':
        $ref: '#/components/responses/404'
      '406':
        $ref: '#/components/responses/406'
      '415':
        $ref: '#/components/responses/415'
      '422':
        $ref: '#/components/responses/422'
      '429':
        $ref: '#/components/responses/429'
/subscriptions/{subscriptionId}:
  parameters:
    - $ref: '#/components/parameters/SubscriptionId'
  get:
    tags:
      - tenantSubscription
    summary: Retrieve information on the existing subscription of a tenant
    description: The GET method is used to retrieve information about this subscription.
    operationId: subscriptionGET
    responses:
      '200':
        description: Successful query of the subscription.
        content:
```

```
            application/json:
              schema:
                oneOf:
                  - $ref: '#/components/schemas/ResourceUsageSubscription'
                  - $ref: '#/components/schemas/SiteResourceUsageSubscription'
        '400':
          $ref: '#/components/responses/400'
        '401':
          $ref: '#/components/responses/401'
        '404':
          $ref: '#/components/responses/404'
        '403':
          $ref: '#/components/responses/403'
        '406':
          $ref: '#/components/responses/406'
        '429':
          $ref: '#/components/responses/429'
    put:
      tags:
        - tenantSubscription
      summary: Modify existing subscription by sending a new data structure
      description: The PUT method is used to update the existing subscription.
      operationId: subscriptionPUT
      requestBody:
        description: new subscription.
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/ResourceUsageSubscription'
                - $ref: '#/components/schemas/SiteResourceUsageSubscription'
        required: true
      responses:
        '200':
          description: Successful update of the subscription.
          content:
            application/json:
              schema:
                oneOf:
                  - $ref: '#/components/schemas/ResourceUsageSubscription'
                  - $ref: '#/components/schemas/SiteResourceUsageSubscription'
        '400':
          $ref: '#/components/responses/400'
        '401':
          $ref: '#/components/responses/401'
        '403':
          $ref: '#/components/responses/403'
        '404':
          $ref: '#/components/responses/404'
        '406':
          $ref: '#/components/responses/406'
        '412':
          $ref: '#/components/responses/412'
        '422':
          $ref: '#/components/responses/422'
        '429':
          $ref: '#/components/responses/429'
    delete:
      tags:
        - tenantSubscription
      summary: Cancel an existing subscription
      description: The DELETE method is used to cancel the existing subscription.
      operationId: subscriptionDELETE
      responses:
        '204':
          description: Successful deletion of the subscription.
        '401':
          $ref: '#/components/responses/401'
        '403':
          $ref: '#/components/responses/403'
        '404':
          $ref: '#/components/responses/404'
        '429':
          $ref: '#/components/responses/429'

components:
  schemas:
    TenantInfo:
```

```
      type: object
      properties:
        customerId:
          description: Identifier of the customer. For the uniqueness of the identifier across the
          MEC system, UUID format is recommended.
          type: string
          format: uuid
          example: 550e8400-e29b-41d4-a716-446655440000
        customerName:
          description: Name of the customer. Examples include the name of a company.
          type: string
          example: company001
        customerCategory:
          description: Category of the customer.
          type: string
          example: Finance, Manufacturing, Retail, Education, Automotive
        tenantId:
          description: Identifier of the tenant. Should be absent in POST requests, and present
          otherwise.
          type: string
          example: tenantId123
        tenantName:
          description: name of the tenant.
          type: string
          example: user001
        resourceUseInfo:
          $ref: '#/components/schemas/ResourceInfo'
        siteList:
          type: array
          items:
            $ref: '#/components/schemas/SiteInfo'
    ResourceQuotaInfo:
      type: object
      properties:
        cpuQuota:
          description: Allowed number of cpus in the MEC system that can be used by the tenant.
          type: integer
          format: int64
          example: 10
        memoryQuota:
          description: Allowed amount of memory (MB) in the MEC system that can be used by the
          tenant.
          type: integer
          format: int64
          example: 10
        diskQuota:
          description: Allowed amount of disk (GB) in the MEC system that can be used by the tenant.
          type: integer
          format: int64
          example: 10
    SiteResourceQuotaInfo:
      type: object
      properties:
        siteId:
          description: Identifier of an edge site. For the uniqueness of the identifier across the
          MEC system, UUID format is recommended.
          type: string
          format: uuid
          example: 550e8400-e29b-41d4-a716-446655440000
        cpuQuota:
          description: Allowed number of CPUs in the edge site that can be used by the tenant.
          type: integer
          format: int64
          example: 10
        memoryQuota:
          description: Allowed amount of memory (MB) in the edge site that can be used by the
          tenant.
          type: integer
          format: int64
          example: 10
        diskQuota:
          description: Allowed amount of disk (GB) in the edge site that can be used by the tenant.
          type: integer
          format: int64
          example: 10
    ResourceUsageSubscription:
      type: object
      properties:
```

```
      subscriptionType:
        description: Should be set to "ResourceUsageSubscription".
        type: string
        example: ResourceUsageSubscription
      callbackReference:
        description: URI exposed by the client on which to receive notifications via HTTP.
        type: string
        format: uri
        example: http://localhost:3000/api/callback
      requestTestNotification:
        description: Set to TRUE by the service consumer to request a test notification via HTTP
        on the callbackReference URI, as specified in ETSI GS MEC 009, clause 6.12a.
        type: boolean
        example: false
      websockNotifConfig:
        $ref: '#/components/schemas/WebsockNotifConfig'
      _links:
        type: object
        properties:
          self:
            $ref: '#/components/schemas/LinkType'
      customerId:
        description: Identifier to uniquely specify the customer for the subscription.
        type: string
        format: uuid
        example: 550e8400-e29b-41d4-a716-446655440000
      tenantId:
        description: Identifier of the tenant.
        type: string
        example: tenantId123
      notificationTrigger:
        description: Set for trigger-based event notification reporting.
        type: object
        properties:
          triggerType:
            description: Trigger for the notification.
            type: string
            enum: ['10', '11', '20', '21', '30', '31']
            example: 10
          threshold:
            description: Threshold for trigger-based event reporting.
            type: integer
            format: uint8
            example: 20
          greaterOrLess:
            description: Indicator for the triggering condition, false means greater than or equal
            to the threshold, true means less than or equal to the threshold.
            type: boolean
            example: false
      expiryDeadline:
        description: The expiration time of the subscription determined by the CSE.
        type: string
        format: date-time
        example: 2017-07-21T17:32:28Z
  SiteResourceUsageSubscription:
    type: object
    properties:
      subscriptionType:
        description: Should be set to "SiteResourceUsageSubscription".
        type: string
        example: SiteResourceUsageSubscription
      callbackReference:
        description: URI exposed by the client on which to receive notifications via HTTP.
        type: string
        format: uri
        example: http://localhost:3000/api/callback
      requestTestNotification:
        description: Set to TRUE by the service consumer to request a test notification via HTTP
        on the callbackReference URI, as specified in ETSI GS MEC 009, clause 6.12a.
        type: boolean
        example: false
      websockNotifConfig:
        $ref: '#/components/schemas/WebsockNotifConfig'
      _links:
        type: object
        properties:
          self:
            $ref: '#/components/schemas/LinkType'
```

```
      customerId:
        description: Identifier to uniquely specify the customer for the subscription.
        type: string
        format: uuid
        example: 550e8400-e29b-41d4-a716-446655440000
      tenantId:
        description: Identifier of the tenant.
        type: string
        example: tenantId123
      siteList:
        description: Identifiers to uniquely specify a list of edge sites for the subscription.
        type: array
        items:
          type: object
          properties:
            siteId:
              type: string
              format: uuid
      notificationTrigger:
        description: Set for trigger-based event notification reporting.
        type: object
        properties:
          triggerType:
            description: Trigger for the notification.
            type: string
            enum: ['10', '11', '20', '21', '30', '31']
            example: 10
          threshold:
            description: Threshold for trigger-based event reporting.
            type: integer
            format: uint8
            example: 20
          greaterOrLess:
            description: Indicator for the triggering condition, false means greater than or equal
            to the threshold, true means less than or equal to the threshold.
            type: boolean
            example: false
      expiryDeadline:
        description: The expiration time of the subscription determined by the CSE.
        type: string
        format: date-time
        example: 2017-07-21T17:32:28Z
  SubscriptionLinkList:
    type: object
    properties:
      _links:
        type: object
        properties:
          self:
            $ref: '#/components/schemas/LinkType'
      subscription:
        type: array
        items:
          type: object
          properties:
            href:
              description: The URI referring to the subscription.
              type: string
              format: uri
              example: http://localhost/cse/v1/subscriptions/subscription01
            subscriptionType:
              description: Type of the subscription. The string should be set according to the
              "subscriptionType" attribute of the associated subscription data type defined.
              type: string
              example: ResourceUsageSubscription
  ResourceUsageNotification:
    type: object
    properties:
      notificationType:
        description: Should be set to "ResourceUsageNotification".
        type: string
        example: ResourceUsageNotification
      timeStamp:
        description: Time stamp.
        type: string
        format: date-time
        example: 2017-07-21T17:32:28Z
      customerId:
```

```
        description: Identifier to uniquely specify the customer for the subscription.
        type: string
        format: uuid
        example: 550e8400-e29b-41d4-a716-446655440000
      tenantId:
        description: Identifier of the tenant.
        type: string
        example: tenantId123
      resourceUseInfo:
        $ref: '#/components/schemas/ResourceUseInfo'
  SiteResourceUsageNotification:
    type: object
    properties:
      notificationType:
        description: Should be set to "SiteResourceUsageNotification".
        type: string
        example: SiteResourceUsageNotification
      timeStamp:
        description: Time stamp.
        type: string
        format: date-time
        example: 2017-07-21T17:32:28Z
      customerId:
        description: Identifier to uniquely specify the customer for the subscription.
        type: string
        format: uuid
        example: 550e8400-e29b-41d4-a716-446655440000
      tenantId:
        description: Identifier of the tenant.
        type: string
        example: tenantId123
      resourceUseInfo:
        type: array
        items:
          type: object
          properties:
            siteId:
              description: Identifier of an edge site.
              type: string
              format: uuid
            siteResourceUseInfo:
              $ref: '#/components/schemas/ResourceUseInfo'
  ExpiryNotification:
    type: object
    properties:
      notificationType:
        description: Should be set to "ExpiryNotification".
        type: string
        example: ExpiryNotification
      _links:
        type: object
        properties:
          subscription:
            description: URI identifying the subscription which has expired.
            $ref: '#/components/schemas/LinkType'
      expiryDeadline:
        description: Time stamp.
        type: string
        format: date-time
        example: 2017-07-21T17:32:28Z
  ResourceUseInfo:
    type: object
    properties:
      cpuUsed:
        description: used number of cpus by the tenant.
        type: integer
        format: int64
        example: 10
      cpuRemain:
        description: remaining number of cpus that can be used by the tenant.
        type: integer
        format: int64
        example: 10
      memoryUsed:
        description: used amount of memory(MB) by the tenant.
        type: integer
        format: int64
        example: 10
```

```
      memoryRemain:
        description: remaining amount of memory(MB) that can be used by the tenant.
        type: integer
        format: int64
        example: 10
      diskUsed:
        description: used amount of disk(GB) by the tenant.
        type: integer
        format: int64
        example: 10
      diskRemain:
        description: remaining amount of disk(GB) that can be used by the tenant.
        type: integer
        format: int64
        example: 10
  ResourceInfo:
    type: object
    properties:
      cpuQuota:
        description: allowed number of cpus that can be used by the tenant.
        type: integer
        format: int64
        example: 10
      cpuUsed:
        description: used number of cpus by the tenant.
        type: integer
        format: int64
        example: 10
      cpuRemain:
        description: remaining number of cpus that can be used by the tenant.
        type: integer
        format: int64
        example: 10
      memoryQuota:
        description: allowed amount of memory(MB) that can be used by the tenant.
        type: integer
        format: int64
        example: 10
      memoryUsed:
        description: used amount of memory(MB) by the tenant.
        type: integer
        format: int64
        example: 10
      memoryRemain:
        description: remaining amount of memory(MB) that can be used by the tenant.
        type: integer
        format: int64
        example: 10
      diskQuota:
        description: allowed amount of disk(GB) that can be used by the tenant.
        type: integer
        format: int64
        example: 10
      diskUsed:
        description: used amount of disk(GB) by the tenant
        type: integer
        format: int64
        example: 10
      diskRemain:
        description: remaining amount of disk(GB) that can be used by the tenant.
        type: integer
        format: int64
        example: 10
  SiteInfo:
    type: object
    properties:
      siteId:
        description: identifier of an edge site. For the uniqueness of the identifier across the
        MEC system, UUID format is recommended.
        type: string
        format: uuid
        example: 550e8400-e29b-41d4-a716-446655440000
      resourceInfo:
        $ref: '#/components/schemas/ResourceInfo'
  WebsockNotifConfig:
    type: object
    properties:
      websocketUri:
```

```
        description: the Websocket URI to be used for delivering notifications.
        type: string
        format: uri
        example: ws://example.net:8080
      requestWebsocketUri:
        description: indicate whether the Websocket delivery is requested.
        type: boolean
  LinkType:
    type: object
    properties:
      href:
        type: string
  ProblemDetails:
    description: The definition of the general "ProblemDetails" data structure from IETF RFC 7807
    is reproduced in this structure. Compared to IETF RFC 7807, the "status" and "detail"
    attributes are mandated to be included by the present document. It is possible that particular
    APIs in the present document, or particular implementations, define extensions to define
    additional attributes that provide more information about the error.
    type: object
    properties:
      type:
        description: A URI reference according to IETF RFC 3986 that identifies the problem type.
        type: string
        format: uri
      title:
        description: A short, human-readable summary of the problem type.
        type: string
      status:
        description: The HTTP status code for this occurrence of the problem.
        type: integer
      detail:
        description: A human-readable explanation specific to this occurrence of the problem.
        type: string
      instance:
        description: A URI reference that identifies the specific occurrence of the problem. It
        may yield further information if dereferenced.
        type: string
        format: uri
parameters:
  TenantId:
    in: path
    name: tenantId
    required: true
    schema:
      type: string
  CustomerId:
    in: query
    name: customerId
    required: true
    schema:
      type: string
      format: uuid
  CustomerName:
    in: query
    name: customerName
    required: true
    schema:
      type: string
  QueryTenantId:
    in: query
    name: tenantId
    required: false
    schema:
      type: string
  TenantName:
    in: query
    name: tenantName
    required: false
    schema:
      type: string
  QuerySiteId:
    in: query
    name: siteId
    required: false
    schema:
      type: string
      format: uuid
  SiteId:
```

```
    in: path
    name: siteId
    required: true
    schema:
      type: string
      format: uuid
  SubscriptionType:
    in: query
    name: subscription_type
    required: false
    schema:
      type: string
  SubscriptionId:
    in: path
    name: subscriptionId
    required: true
    schema:
      type: string
responses:
  400:
    description: It is used to indicate that incorrect parameters were passed to the request.
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ProblemDetails'
  401:
    description: It is used when the client did not submit credentials.
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ProblemDetails'
  403:
    description: The operation is not allowed given the current status of the resource.
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ProblemDetails'
  404:
    description: It is used when a client provided a URI that cannot be mapped to a valid resource
    URI.
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ProblemDetails'
  406:
    description: It is used to indicate that the server cannot provide the any of the content
    formats supported by the client.
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ProblemDetails'
  412:
    description: It is used when a condition has failed during conditional requests, e.g. when
    using ETags to avoid write conflicts.
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ProblemDetails'
  414:
    description: It is used to indicate that the server is refusing to process the request because
    the request URI is longer than the server is willing or able to process.
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ProblemDetails'
  415:
    description: It is used to indicate that the server or the client does not support the content
    type of the entity body.
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ProblemDetails'
  422:
    description: 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.
    content:
      application/json:
```

```
        schema:
          $ref: '#/components/schemas/ProblemDetails'
    429:
      description: It is used when a rate limiter has triggered.
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
```

Annex A (informative): Complementary material for API utilization

To complement the definitions for each method and resource defined in the interface clauses of the present document, ETSI MEC ISG is providing for the user application lifecycle management API a supplementary description file compliant to the OpenAPI Specification [i.3].

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

The supplementary description file, relating to the present document, is located at:

- https://forge.etsi.org/rep/mec/gs048-cse-api.

Annex B (informative): Change History

Date	Version	Information about changes
August 2023	3.0.1	Initial draft
November 2023	3.0.2	Updated with the agreements in MEC(23)000350r2, MEC(23)000447r1, MEC(23)000351r1, MEC(23)000448r1, MEC(23)000352r1, and MEC(23)000449r2.
November 2023	3.0.3	Updated with the agreements in MEC(23)000353.
February 2024	3.0.4	Updated with the agreements in MEC(24)000071, MEC(24)000072, MEC(24)000073, MEC(24)000082, MEC(24)000083 and MEC(24)000084.
February 2024	3.0.5	Final draft V3.0.5 is similar to Stable draft V3.0.4, and is ready to be sent to MEC RC for review.
March 2024	3.0.6	Updated with the agreements in MEC(24)000121, MEC(24)000122 and MEC(24)000124, which are to address the comments raised during the RC for review. Other updates to address the comments: - added informative Annex "Complementary material for API utilization" - fixed the bug in 6.1 Introduction
April 2024	3.1.1	First published version.

History

Document history		
V3.1.1	April 2024	Publication