

ETSI GS MEC 015 V3.1.1 (2024-04)



Multi-access Edge Computing (MEC); Traffic Management APIs

Disclaimer

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

Reference

RGS/MEC-0015v311TrafMngtAPIs

Keywords

API, management, MEC, QoS, traffic

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/CommitteeSupportStaff.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.....	8
3 Definition of terms, symbols and abbreviations.....	8
3.1 Terms.....	8
3.2 Symbols.....	8
3.3 Abbreviations	8
4 Void.....	8
5 Overview	9
6 Description of the service (informative).....	9
6.1 Introduction	9
6.2 Sequence diagrams	10
6.2.1 General.....	10
6.2.2 Register to Bandwidth Management Service	10
6.2.3 Unregister from Bandwidth Management Service.....	11
6.2.4 Update requested bandwidth requirements on BWM Service	11
6.2.5 Get configured bandwidth allocation from BWM Service	12
6.2.6 REST based subscribe-notify model.....	12
6.2.6.1 Subscribing to BWM information event notifications	12
6.2.6.2 Receiving notification of BWM information event subscription	13
6.2.6.3 Updating subscription for BWM information event notifications.....	13
6.2.6.4 Unsubscribing from BWM information event notifications.....	14
6.2.7 Get MTS service Info from the MTS Service.....	15
6.2.8 Register to the MTS service.....	15
6.2.9 Unregister from the MTS service	16
6.2.10 Update requested requirements on the MTS service.....	16
6.2.11 Get configured MTS session from the MTS service.....	17
7 Data Model	17
7.1 Introduction	17
7.2 Resource data types	17
7.2.1 Introduction.....	17
7.2.2 Type: BwInfo.....	18
7.2.3 Type: BwInfoDeltas.....	18
7.2.4 Type: MtsCapabilityInfo.....	20
7.2.5 Type: MtsSessionInfo.....	20
7.3 Subscription data types.....	22
7.3.1 Introduction.....	22
7.3.2 Type: BwChgEventSubscription	22
7.3.3 Type: SubscriptionLinkList	22
7.4 Notification data types.....	23
7.4.1 Introduction.....	23
7.4.2 Type: BwChgEventNotification	23
7.5 Referenced structure data types.....	23
7.5.1 Introduction.....	23
7.5.2 Type: LinkType	24
7.5.3 Type: WebsocketNotifConfig.....	24
8 BWM API definition.....	24
8.1 Introduction	24

8.2	Global definitions and resource structure	24
8.3	Resource: individual bandwidthAllocation	25
8.3.1	Description.....	25
8.3.2	Resource definition	25
8.3.3	Resource Methods	26
8.3.3.1	GET.....	26
8.3.3.2	PUT	26
8.3.3.3	PATCH	27
8.3.3.4	POST.....	28
8.3.3.5	DELETE	28
8.4	Resource: a list of bandwidthAllocations	29
8.4.1	Description.....	29
8.4.2	Resource definition	29
8.4.3	Resource Methods	29
8.4.3.1	GET.....	29
8.4.3.2	PUT	30
8.4.3.3	PATCH	30
8.4.3.4	POST.....	30
8.4.3.5	DELETE	31
8.5	Resource: subscriptions	31
8.5.1	Description.....	31
8.5.2	Resource definition	31
8.5.3	Resource methods.....	32
8.5.3.1	GET.....	32
8.5.3.2	PUT	33
8.5.3.3	PATCH	33
8.5.3.4	POST.....	33
8.5.3.5	DELETE	34
8.6	Resource: existing subscription.....	34
8.6.1	Description.....	34
8.6.2	Resource definition	34
8.6.3	Resource methods.....	34
8.6.3.1	GET.....	34
8.6.3.2	PUT	35
8.6.3.3	PATCH	37
8.6.3.4	POST.....	37
8.6.3.5	DELETE	37
8.7	Resource: Notification callback	38
8.7.1	Description.....	38
8.7.2	Resource definition	38
8.7.3	Resource methods.....	38
8.7.3.1	GET.....	38
8.7.3.2	PUT	38
8.7.3.3	PATCH	39
8.7.3.4	POST.....	39
8.7.3.5	DELETE	39
9	MTS API definition.....	40
9.1	Introduction	40
9.2	Global definitions and resource structure	40
9.3	Resource: MTS information	41
9.3.1	Description.....	41
9.3.2	Resource definition	41
9.3.3	Resource Methods	41
9.3.3.1	GET.....	41
9.4	Resource: individual MTS session	42
9.4.1	Description.....	42
9.4.2	Resource definition	42
9.4.3	Resource Methods	42
9.4.3.1	GET.....	42
9.4.3.2	PUT	43
9.4.3.3	DELETE	44

9.5	Resource: a list of MTS sessions	45
9.5.1	Description	45
9.5.2	Resource definition	45
9.5.3	Resource Methods	45
9.5.3.1	GET	45
9.5.3.2	POST	46
Annex A (informative):	Complementary material for API utilization	48
History		49

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 focuses on the Traffic Management (TM) MEC service. It describes the TM related information including access control, information flows, required information and operations. The present document specifies the necessary API with the data model and data format.

2 References

2.1 Normative references

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

Referenced documents which are not found to be publicly available in the expected location might be found 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] Void.
- [2] Void.
- [3] Void.
- [4] Void.
- [5] Void.
- [6] [ETSI GS MEC 009](#): "Multi-access Edge Computing (MEC); General principles, patterns and common aspects of MEC Service APIs".
- [7] [IETF RFC 7396](#): "JSON Merge Patch".
- [8] [IEEE 802.11TM-2016](#): "IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications".
- [9] Void.
- [10] [IETF RFC 1166](#): "Internet numbers".
- [11] [IETF RFC 5952](#): "A Recommendation for IPv6 Address Text Representation".
- [12] [IETF RFC 4632](#): "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan".

2.2 Informative references

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

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

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

- [i.1] ETSI GS MEC 002: "Multi-access Edge Computing (MEC); Use Cases and Requirements".
- [i.2] [OpenAPI™ Specification](#).
- [i.3] ETSI GR MEC 001: "Multi-access Edge Computing (MEC); Terminology".
- [i.4] ETSI TS 123 288 (V17.9.0): "5G; Architecture enhancements for 5G System (5GS) to support network data analytics services (3GPP TS 23.288 version 17.9.0 Release 17)".
- [i.5] ETSI TS 123 501: "5G; System architecture for the 5G System (5GS) (3GPP TS 23.501 Release 17)".

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.3] apply.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GR MEC 001 [i.3] and the following apply:

ATSSS	Access Traffic Steering, Switching, Splitting
BW	BandWidth
BWM	BandWidth Management
BWMS	BandWidth Management Service
CDN	Content Delivery Network
DSCP	Differentiated Services Code Point
MTS	Multi-access Traffic Steering
NR	New Radio
NWDAF	NetWork Data Analytical Function
OAI	Open API Initiative
RTT	Round Trip Time
TM	Traffic Management
TMS	Traffic Management Service
UTRA	Universal Terrestrial Radio Access

4 Void

5 Overview

The present document specifies the Traffic Management (TM) APIs to support the requirements defined for Multi-Access Edge Computing in ETSI GS MEC 002 [i.1]. There are two TM services: BandWidth Management (BWM) service and Multi-access Traffic Steering (MTS) service. Clause 6 introduces how TM services can be used by the multi-access edge applications and by the multi-access edge platform. It describes the information flows used for TM services.

The information that can be exchanged over the TM APIs is described in clause 7 which provides detailed description on all information elements that are used for TM services.

Clauses 8 and 9 describe the actual TM APIs (BWM API and MTS API) providing detailed information on how information elements are mapped into a RESTful API design.

Figure 5-1 illustrates the mission of the TM services, which may optionally run as part of the platform or as an application. Different applications, whether managing a single instance or several sessions (for example CDN), may request specific Bandwidth Management (BWM) or/and Multi-access Traffic Steering (MTS) requirements for the whole application instance or different requirements per session. The TM services can aggregate all the requests and act in a manner that will help optimize the BW usage and improve Quality of user Experience for applications.

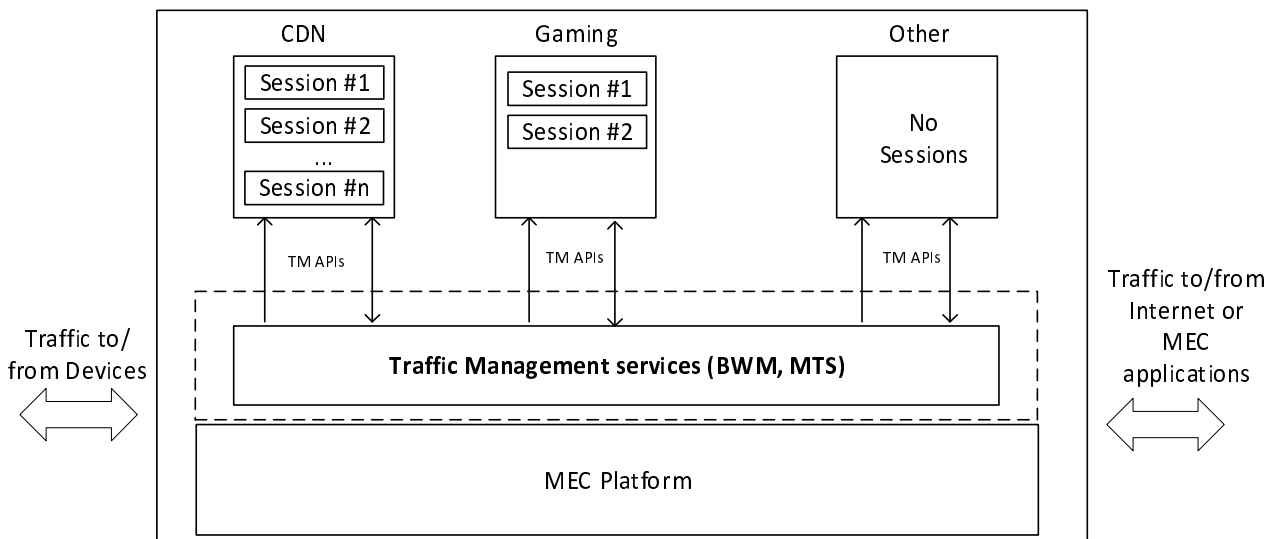


Figure 5-1: Traffic Management services description

6 Description of the service (informative)

6.1 Introduction

Different MEC applications running in parallel on the same MEC host may require specific static/dynamic up/down bandwidth resources, including bandwidth size and bandwidth priority. In some cases, different sessions running in parallel on the same application may each have specific bandwidth requirements. In addition, sessions driven by applications running from closer to end user (shorter RTT) may receive unfair advantage over sessions driven by applications running from distant locations (outside the RAN). To resolve potential resource conflicts between such competing applications, the following optional traffic management services may be used:

- BandWidth Management (BWM) service; and
- Multi-access Traffic Steering (MTS) service.

The BWM service is for allocating/adjusting bandwidth resources, including bandwidth size and bandwidth priority, for MEC applications, and allows MEC applications to provide bandwidth requirements.

The MTS service is for seamlessly steering/splitting/duplicating application data traffic across multiple access network connections. The MTS allows:

- 1) MEC applications to get informed of various MTS capabilities and multi-access network connection info.
- 2) MEC applications to provide requirements, e.g. delay, throughput, loss, for influencing traffic management operations.

The specific session or MEC application will be identified using a set of filters within the resource request.

6.2 Sequence diagrams

6.2.1 General

The following clauses describe how multi-access edge applications can use TMS to update/receive Bandwidth Management (BWM) or/and Multi-access Traffic Steering (MTS) information to/from the MEC platform. The sequence diagrams that are relevant for TMS are presented.

The TM APIs enable the MEC applications to register or unregister for specific bandwidth allocation or/and multi-access traffic steering requirement. The "Registration" flow is used to create a bandwidthAllocation as shown in clause 6.2.2 or a mtsSession as shown in clause 6.2.7. It is operated on per-allocation/session basis, and can be used multiple times by the application to create multiple bandwidthAllocations or mtsSessions. The "Unregistration" flow is used to delete a bandwidthAllocation as shown in clause 6.2.3 or a mtsSession as shown in clause 6.2.8.

The present document of TM APIs contains the HTTP protocol bindings for traffic management functionality using the REST architectural style.

6.2.2 Register to Bandwidth Management Service

Figure 6.2.2-1 shows a scenario where a MEC Application instance registers to BWMS.

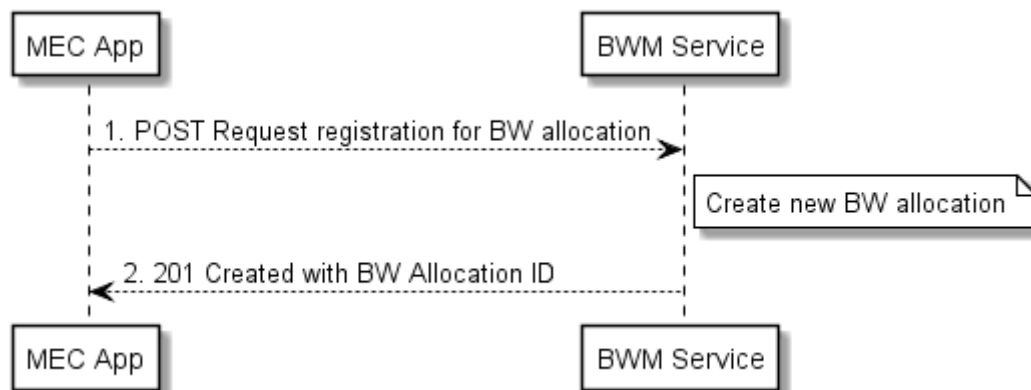


Figure 6.2.2-1: Flow of MEC Application registration to BWMS

MEC Application instance registration to BWMS, as illustrated in figure 6.2.2-1, consists of the following steps:

- 1) MEC application instance sends a request to register to the BWMS with the requested bandwidth requirements (bandwidth size/priority).
- 2) BWMS responds with a registration and initialization approval.

6.2.3 Unregister from Bandwidth Management Service

Figure 6.2.3-1 shows a scenario where a MEC Application Instance unregisters from BWMS.

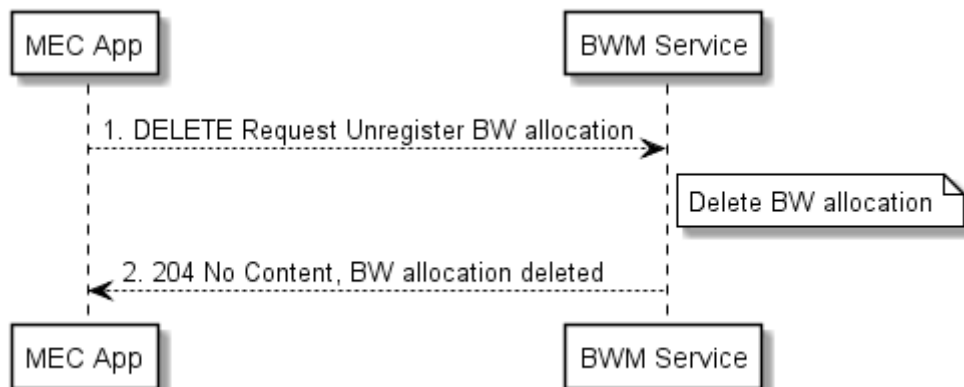


Figure 6.2.3-1: Flow of MEC Application unregistering BW allocation from BWMS

MEC Application Instance unregistering from BWMS, as illustrated in figure 6.2.3-1, consists of the following steps:

- 1) MEC Application instance sends an unregister request to BWMS.
- 2) BWMS responds with an unregistration approval.

6.2.4 Update requested bandwidth requirements on BWM Service

Figure 6.2.4-1 shows a scenario where a MEC Application instance updates its requested bandwidth requirements on the BWMS.

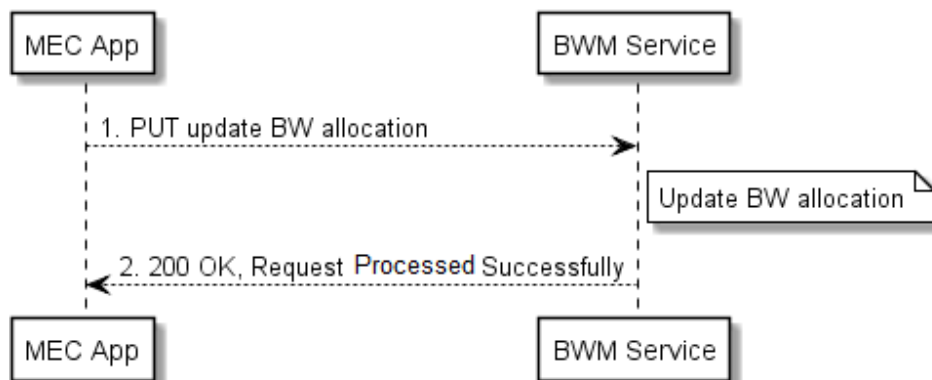


Figure 6.2.4-1: Flow of MEC application updating its requested bandwidth requirements on BWMS

MEC application instance updating its requested bandwidth requirements on BWMS, as illustrated in figure 6.2.4-1, consists of the following steps:

- 1) MEC Application instance sends a request to update a specific bandwidth allocation on the BWMS.
- 2) BWMS responds with an update approval.

6.2.5 Get configured bandwidth allocation from BWM Service

Figure 6.2.5-1 shows a scenario where a MEC Application instance gets its configured bandwidth allocation from the BWMS.

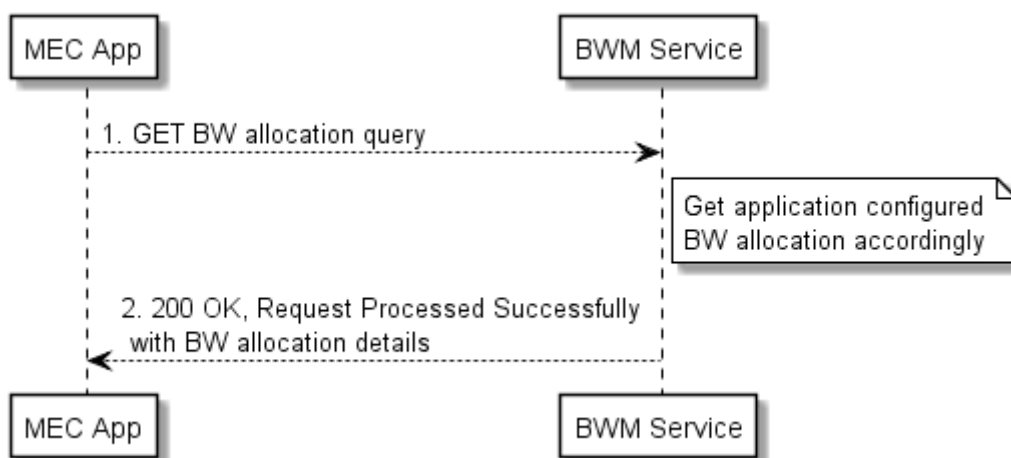


Figure 6.2.5-1: Flow of MEC Application getting its configured bandwidth allocation from BWMS

MEC Application instance gets its configured bandwidth from BWMS, as illustrated in figure 6.2.5-1, consists of the following steps:

- 1) MEC Application instance sends a request to get its configured bandwidth allocation on the BWMS.
- 2) BWMS responds with the BW allocation details.

6.2.6 REST based subscribe-notify model

6.2.6.1 Subscribing to BWM information event notifications

To receive notifications on selected BWM information event, the service consumer creates a subscription to certain BW information change event that is available at BWM service. Figure 6.2.6.1-1 shows a scenario where the service consumer uses REST based procedures to create a subscription for BWM information event notification.

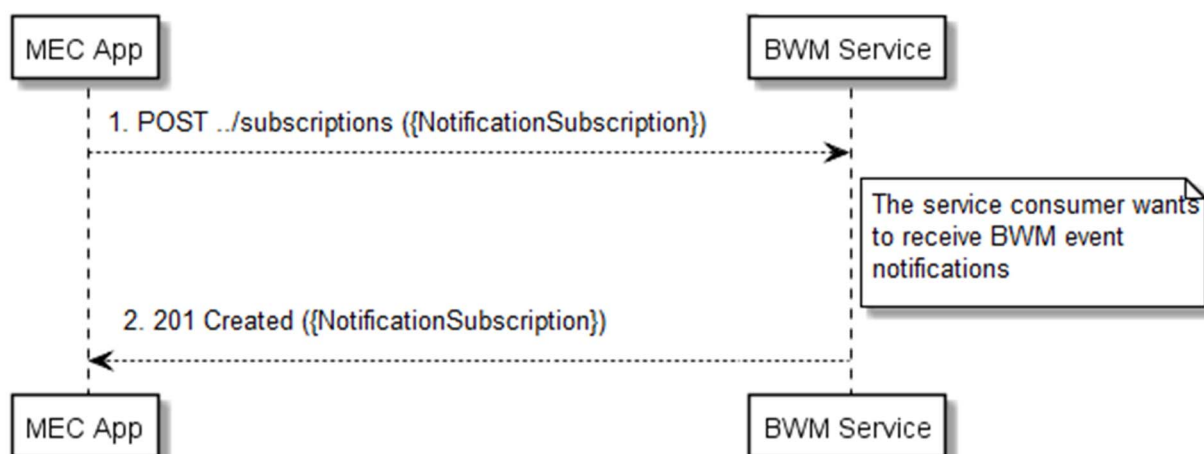


Figure 6.2.6.1-1: Flow of subscribing to BWM information event notifications

Subscribing to the BWM information event notifications, as illustrated in figure 6.2.6.1-1, consists of the following steps.

When service consumer wants to receive notification about the BWM information event, it creates a subscription to the BWM information event notifications:

- 1) The service consumer sends a POST request with the message body containing the {NotificationSubscription} data structure to the resource representing BWM information subscription. The variable {NotificationSubscription} is replaced with the data type specified for different BWM information event subscriptions, and defines the subscribed event, the filtering criteria and the address where the service consumer wishes to receive the BWM information event notification.
- 2) BWM service sends "201 Created" response with the message body containing the data structure specific to that of BWM information event subscription. The data structure contains the address of the resource created and the subscribed BWM information event type. The address of the resource created is also contained in the message header.

6.2.6.2 Receiving notification of BWM information event subscription

Figure 6.2.6.2-1 presents the scenario where BWM service sends BWM information event notification to the service consumer (MEC application or a MEC platform) about the BW change event information.

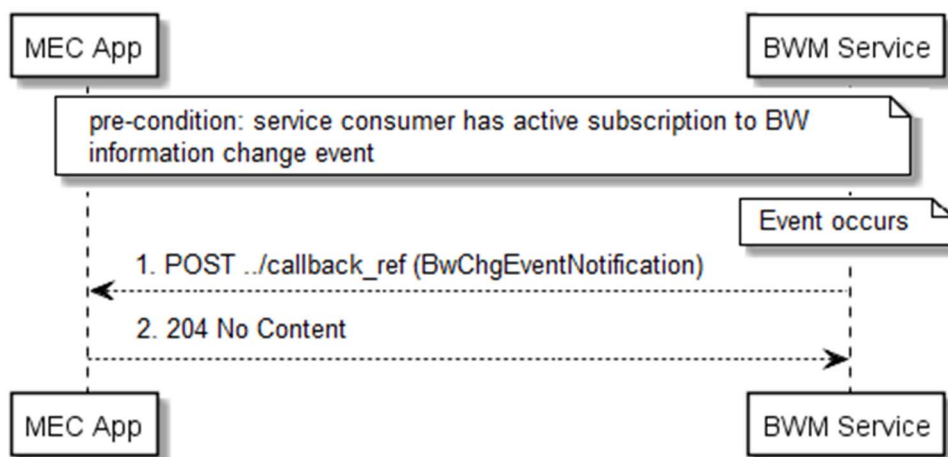


Figure 6.2.6.2-1: Flow of BWM information event notification

BWM service sends a notification to the subscribed service consumer as illustrated in figure 6.2.6.2-1, with the following steps:

- 1) BWM service sends a POST request with the message body containing the BwChgEventNotification data structure to the callback reference address included by the service consumer in the BwChgEventSubscription event subscription.
- 2) Service consumer sends a "204 No Content" response to BWM service.

6.2.6.3 Updating subscription for BWM information event notifications

Figure 6.2.6.3-1 shows a scenario where the service consumer needs to update an existing subscription for a BWM information event notification. The subscription update is triggered e.g. by the need to change the existing subscription, or due to the expiry of the notification.

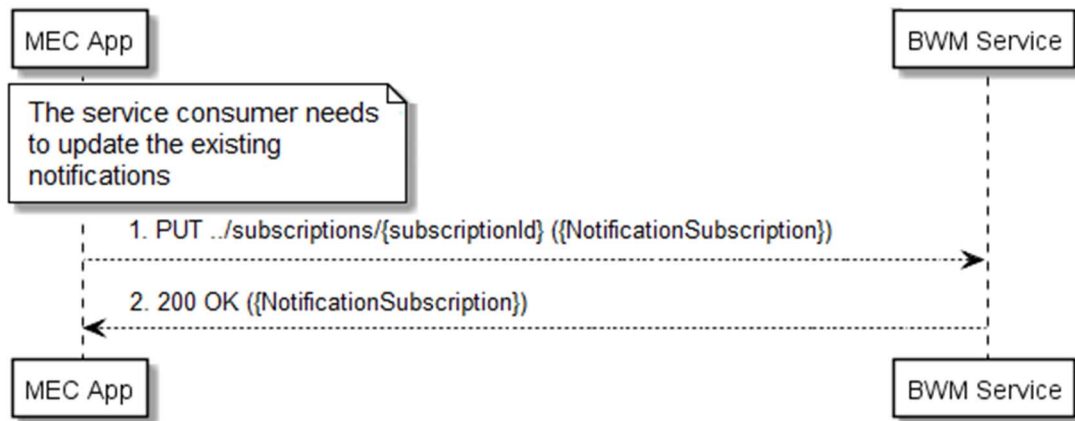


Figure 6.2.6.3-1: Flow of service consumer updating subscription for BWM information event notifications

Updating the subscription for BWM information event notification, as illustrated in figure 6.2.6.3-1, consists of the following steps.

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

- 1) Service consumer updates the subscription resource by sending a PUT request to the resource representing the BWM information event subscription that was created with the modified data structure of that BWM information event subscription.
- 2) BWM service returns "200 OK" with the message body containing the accepted data structure specific to that BWM information event subscription.

6.2.6.4 Unsubscribing from BWM information event notifications

When the service consumer does not want to receive notifications anymore after subscribing to BWM information events, the service consumer unsubscribes from the BWM information event notifications. Figure 6.2.6.4-1 shows a scenario where the service consumer uses REST based procedures to delete the subscription for BWM information event notifications.

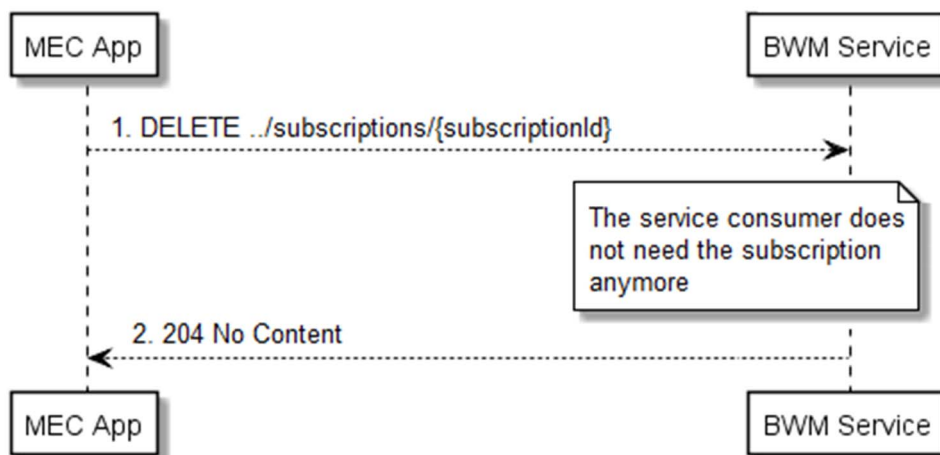


Figure 6.2.6.4-1: Flow of unsubscribing for BWM information event notifications

Unsubscribing from the BWM information event notification, as illustrated in figure 6.2.6.4-1, consists of the following steps.

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

- 1) Service consumer sends a DELETE request to the resource representing the BWM information event subscription that was created.
- 2) BWM service sends "204 No Content" response.

6.2.7 Get MTS service Info from the MTS Service

Figure 6.2.7-1 shows a scenario where a MEC Application instance gets the available MTS service information from the MTS service.

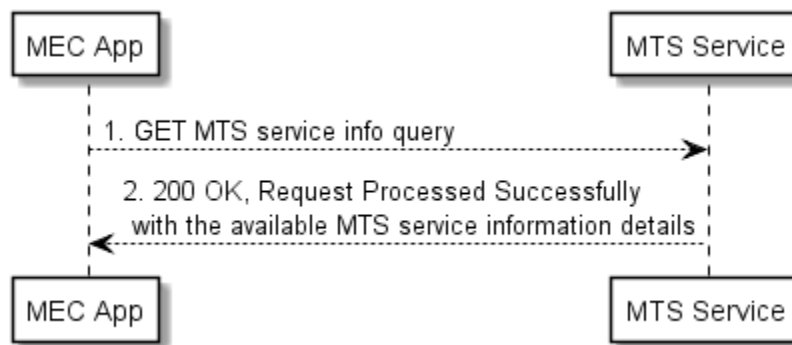


Figure 6.2.7-1: Flow of MEC Application getting the MTS service info

MEC Application instance gets the available MTS service info from the MTS service, as illustrated in figure 6.2.7-1, consists of the following steps:

- 1) MEC Application instance sends a request to get the available MTS service information.
- 2) The MTS service responds with the available MTS service information details.

6.2.8 Register to the MTS service

Figure 6.2.8-1 shows a scenario where a MEC Application instance registers to the MTS service.

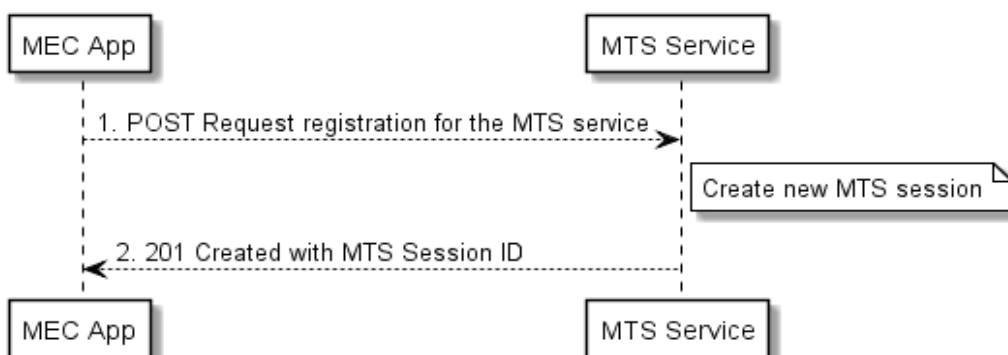


Figure 6.2.8-1: Flow of MEC Application registration to the MTS service

MEC Application instance registration to the MTS service, as illustrated in figure 6.2.8-1, consists of the following steps:

- 1) MEC Application instance sends a request to register to the MTS service with the requested requirements.
- 2) The MTS service responds with a registration and initialization approval.

6.2.9 Unregister from the MTS service

Figure 6.2.9-1 shows a scenario where a MEC Application instance unregisters from the MTS service.

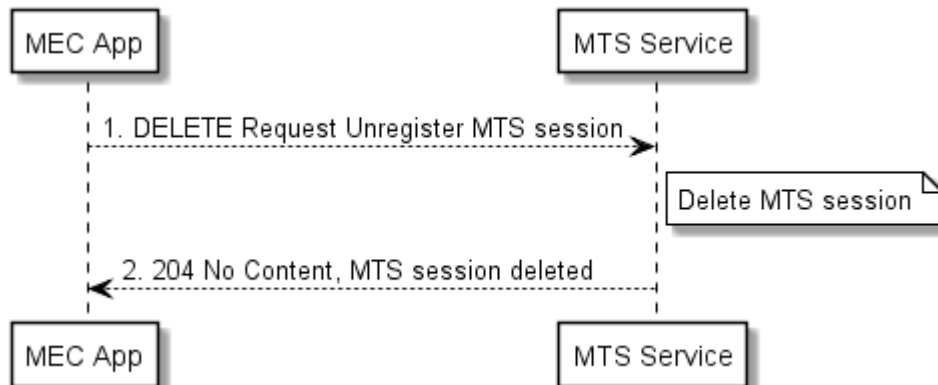


Figure 6.2.9-1: Flow of MEC Application unregistering MTS session from the MTS service

MEC Application instance unregistering from the MTS service, as illustrated in figure 6.2.9-1, consists of the following steps:

- 1) MEC Application instance sends an unregister request to the MTS service.
- 2) MTS responds with an unregistration approval.

6.2.10 Update requested requirements on the MTS service

Figure 6.2.10-1 shows a scenario where a MEC Application instance updates its requested requirements on the MTS service.

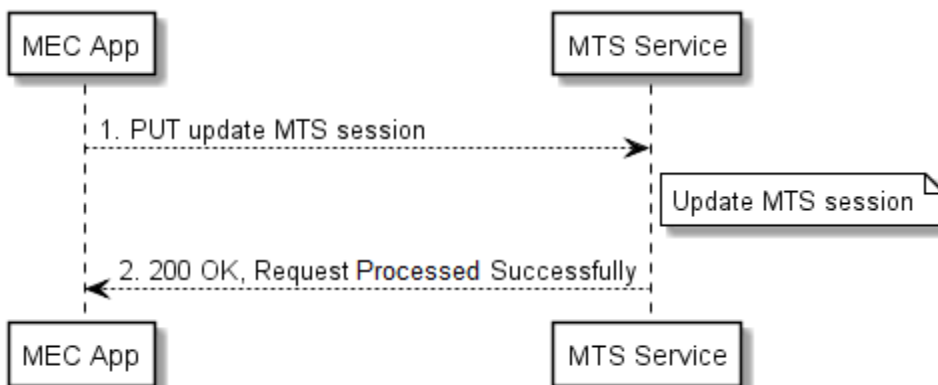


Figure 6.2.10-1: Flow of MEC application updating its requested requirements on the MTS service

MEC Application instance updating its requested requirements on the MTS service, as illustrated in figure 6.2.10-1, consists of the following steps:

- 1) MEC Application instance sends a request to update a specific MTS session on the MTS service.
- 2) The MTS service responds with an update approval.

6.2.11 Get configured MTS session from the MTS service

Figure 6.2.11-1 shows a scenario where a MEC Application instance gets its configured MTS session from the MTS service.

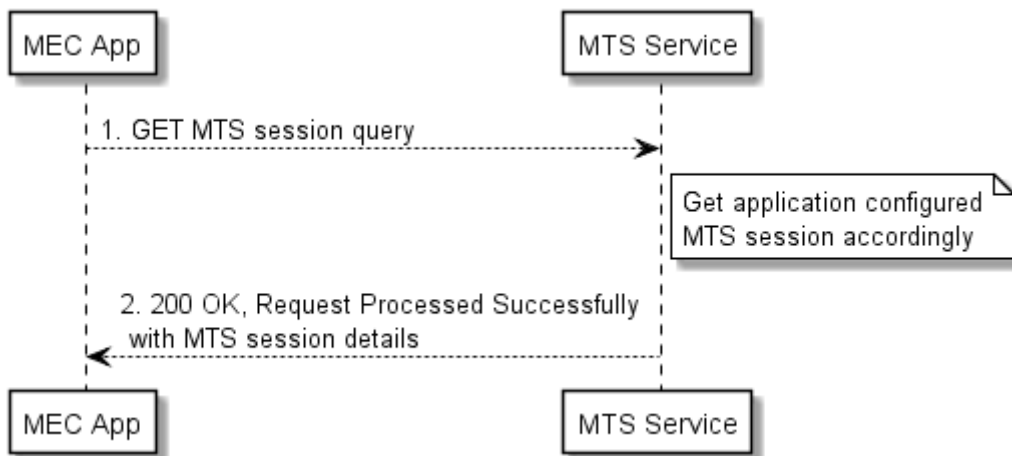


Figure 6.2.11-1: Flow of MEC Application getting its configured MTS session info from the MTS service

MEC Application instance gets its configured MTS session information from the MTS service, as illustrated in figure 6.2.11-1, consists of the following steps:

- 1) MEC Application instance sends a request to get its configured MTS session information on the MTS service.
- 2) The MTS service responds with the MTS session details.

7 Data Model

7.1 Introduction

The following clauses provide the description of the Data Model.

7.2 Resource data types

7.2.1 Introduction

This clause defines data structures to be used in resource representations.

7.2.2 Type: BwInfo

Table 7.2.2-1: Elements of BwInfo

Element	Type	Cardinality	Description
allocationId	String	0..1	Bandwidth allocation instance identifier.
timeStamp	Structure (inlined)	0..1	Time stamp to indicate when the corresponding information elements are sent.
>seconds	Uint32	1	The seconds part of the Time. Time is defined as Unix-time since January 1, 1970, 00:00:00 UTC.
>nanoSeconds	Uint32	1	The nanoseconds part of the Time. Time is defined as Unix-time since January 1, 1970, 00:00:00 UTC.
appInstId	String	1	Application instance identifier.
appName	String	0..1	Name of the application.
requestType	Enum (inlined)	1	Numeric value (0 to 255) corresponding to specific type of consumer as following: 0 = APPLICATION_SPECIFIC_BW_ALLOCATION. 1 = SESSION_SPECIFIC_BW_ALLOCATION.
sessionFilter	Structure (inlined)	0..N	Session filtering criteria, applicable when requestType is set as SESSION_SPECIFIC_BW_ALLOCATION. Any filtering criteria shall define a single session only. In case multiple sessions match sessionFilter, the request shall be rejected.
>sourceAddress	String	0..1	Source address identity of session. The string for an IPv4 address shall be formatted in the "dotted decimal" notation as defined in IETF RFC 1166 [10]. The string for an IPv6 address shall be formatted according to clause 4 of IETF RFC 5952 [11], with in CIDR notation IETF RFC 4632 [12] used to provide the routing prefix.
>sourcePort	String	0..1	Source port identity of session.
>dstAddress	String	0..1	Destination address identity of session. The string for an IPv4 address shall be formatted in the "dotted decimal" notation as defined in IETF RFC 1166 [10]. The string for an IPv6 address shall be formatted according to clause 4 of IETF RFC 5952 [11], with in CIDR notation IETF RFC 4632 [12] used to provide the routing prefix.
>dstPort	String	0..1	Destination port identity of session.
>protocol	String	0..1	Protocol number.
fixedBWPriority	Enum	0..1	Indicates the allocation priority when dealing with several applications or sessions in parallel. Values are not defined in the present document.
fixedAllocation	String	1	Size of requested fixed BW allocation in [bps].
allocationDirection	String	1	The direction of the requested BW allocation: 00 = Downlink (towards the UE). 01 = Uplink (towards the application/session). 10 = Symmetrical.

7.2.3 Type: BwInfoDeltas

Conforming to JSON merge patch format and processing rules specified IETF RFC 7396 [7], this type represents the attributes whose value are allowed to be updated with HTTP PATCH method in content format JSON. It shall follow the indications provided in table 7.2.3-1.

Table 7.2.3-1: Elements of BwInfoDeltas

Attribute name	Data type	Cardinality	Description
allocationId	String	1	Bandwidth allocation instance identifier
appInsId	String	1	Application instance identifier.
requestType	Enum (inlined)	1	Numeric value (0 to 255) corresponding to specific type of consumer as following: 0 = APPLICATION_SPECIFIC_BW_ALLOCATION. 1 = SESSION_SPECIFIC_BW_ALLOCATION.
sessionFilter	Structure (inlined)	0..N	Session filtering criteria, applicable when requestType is set as SESSION_SPECIFIC_BW_ALLOCATION. Any filtering criteria shall define a single session only. In case multiple sessions match sessionFilter, the request shall be rejected.
>sourceAddress	String	0..1	Source address identity of session. The string for an IPv4 address shall be formatted in the "dotted decimal" notation as defined in IETF RFC 1166 [10]. The string for an IPv6 address shall be formatted according to clause 4 of IETF RFC 5952 [11], with in CIDR notation IETF RFC 4632 [12] used to provide the routing prefix.
>sourcePort	String	0..1	Source port identity of session.
>dstAddress	String	0..1	Destination address identity of session. The string for an IPv4 address shall be formatted in the "dotted decimal" notation as defined in IETF RFC 1166 [10]. The string for an IPv6 address shall be formatted according to clause 4 of IETF RFC 5952 [11], with in CIDR notation IETF RFC 4632 [12] used to provide the routing prefix.
>dstPort	String	0..1	Destination port identity of session.
>protocol	String	0..1	Protocol number.
fixedBWPriority	Enum (inlined)	0..1	Indicates the allocation priority when dealing with several applications or sessions in parallel. Values are not defined in the present document.
fixedAllocation	String	0..1	Size of requested fixed BW allocation in [bps].
allocationDirection	String	0..1	The direction of the requested BW allocation: 00 = Downlink (towards the UE). 01 = Uplink (towards the application/session). 10 = Symmetrical.

7.2.4 Type: MtsCapabilityInfo

Table 7.2.4-1: Attributes of the MtsCapabilityInfo

Attribute name	Data type	Cardinality	Description
timeStamp	Structure (inlined)	0..1	Time stamp to indicate when the corresponding information elements are sent.
>seconds	Uint32	1	Time in seconds in Unix-time since January 1, 1970, 00:00:00 UTC.
>nanoSeconds	Uint32	1	Time in nanoseconds in Unix-time since January 1, 1970, 00:00:00 UTC.
mtsAccessInfo	Structure (inlined)	1..N	The information on access network connection as defined below.
>accessId	Uint32	1	Unique identifier for the access network connection.
>accessType	Uint32	1	Numeric value (0 to 255) corresponding to specific type of access network as following: 0 = Unknown. 1 = Any IEEE802.11-based WLAN technology. 2 = Any 3GPP-based Cellular technology. 3 = Any Fixed Access. 11 = IEEE802.11 a/b/g WLAN. 12 = IEEE 802.11 a/b/g/n WLAN. 13 = IEEE 802.11 a/b/g/n/ac WLAN. 14 = IEEE 802.11 a/b/g/n/ac/ax WLAN (Wi-Fi 6). 15 = IEEE 802.11 b/g/n WLAN. 31 = 3GPP GERAN/UTRA (2G/3G). 32 = 3GPP E-UTRA (4G/LTE). 33 = 3GPP NR (5G).
>metered	Uint32	1	Numeric value (0 to 255) corresponding to the following: 0: the connection is not metered (see note). 1: the connection is metered. 2: unknown.
mtsMode	Uint32	1..N	Numeric value corresponding to a specific MTS operation supported by the TMS: 0 = low cost, i.e. using the unmetered access network connection whenever it is available. 1 = low latency, i.e. using the access network connection with lower latency. 2 = high throughput, i.e. using the access network connection with higher throughput, or/and multiple access network connection simultaneously if supported. 3 = redundancy, i.e. sending duplicated (redundancy) packets over multiple access network connections for high-reliability and low-latency applications. 4 = QoS, i.e. performing MTS based on the specific QoS requirements from the app.
NOTE: A metered connection is a network connection that has a maximum data usage in a specific period, e.g. per hour/day/week/month. The user may get billed extra charges if they go over the allotted amount.			

7.2.5 Type: MtsSessionInfo

Table 7.2.5-1: Elements of MtsSessionInfo

Element	Type	Cardinality	Description
sessionId	String	0..1	MTS session instance identifier.
timeStamp	Structure (inlined)	0..1	Time stamp to indicate when the corresponding information elements are sent.
>seconds	Uint32	1	The seconds part of the Time. Time is defined as Unix-time since January 1, 1970, 00:00:00 UTC.
>nanoSeconds	Uint32	1	The nanoseconds part of the Time. Time is defined as Unix-time since January 1, 1970, 00:00:00 UTC.
appInstId	String	1	Application instance identifier.
appName	String	0..1	Name of the application.

Element	Type	Cardinality	Description
requestType	Enum (inlined)	1	Numeric value (0 to 255) corresponding to specific type of consumer as following: 0 = APPLICATION_SPECIFIC_MTS_SESSION. 1 = FLOW_SPECIFIC_MTS_SESSION.
flowFilter	Structure (inlined)	1..N	Traffic flow filtering criteria, applicable only if when requestType is set as FLOW_SPECIFIC_MTS_SESSION. Any filtering criteria shall define a single session only. In case multiple sessions match flowFilter, the request shall be rejected. If the flowFilter field is included, at least one of its subfields shall be included. Any flowFilter subfield that is not included shall be ignored in traffic flow filtering.
>sourceAddress	String	0..1	Source address identity of session. The string for an IPv4 address shall be formatted in the "dotted decimal" notation as defined in IETF RFC 1166 [10]. The string for an IPv6 address shall be formatted according to clause 4 of IETF RFC 5952 [11], with in CIDR notation IETF RFC 4632 [12] used to provide the routing prefix.
>sourcePort	String	0..1	Source port identity of session.
>dstAddress	String	0..1	Destination address identity of session. The string for an IPv4 address shall be formatted in the "dotted decimal" notation as defined in IETF RFC 1166 [10]. The string for an IPv6 address shall be formatted according to clause 4 of IETF RFC 5952 [11], with in CIDR notation IETF RFC 4632 [12] used to provide the routing prefix.
>dstPort	String	0..1	Destination port identity of session.
>protocol	String	0..1	Protocol number.
>maxSessions	Uint32	0..1	It indicates the maximum number of sessions MTS service will track for the given (sourceIp, dstIp etc.) provided by a service consumer. This attribute shall not be present in the request, but may be present in the response.
>dscp	Uint32	0..1	DSCP in the IPv4 header or Traffic Class in the IPv6 header.
>flowLabel	Uint32	0..1	Flow Label in the IPv6 header, applicable only if the flow is IPv6.
qoS	Structure	1	QoS requirement description of the MTS session, applicable only if mtsMode = 4 (QoS). If the qos field is included, at least one of its subfields shall be included. Any qos subfield that is not included shall be ignored in Multi-access Traffic Steering (MTS).
>minTpt	Unit32	0..1	Minimal throughput in [kbps].
>maxLatency	Unit32	0..1	Tolerable (one-way) delay in [10 nanoseconds].
>maxLoss	Unit32	0..1	Tolerable packet loss rate in [1/10^x].
>maxJitter	Unit32	0..1	Tolerable jitter in [10 nanoseconds].
>priority	Unit32	0..1	numeric value (0 to 255) corresponding to the traffic priority: 0: low; 1: medium; 2: high; 3: critical.
mtsMode	Unit32	1	Numeric value (0 to 255) corresponding to a specific MTS mode of the MTS session: 0 = low cost, i.e. using the unmetered access network connection whenever it is available. 1 = low latency, i.e. using the access network connection with lower latency. 2 = high throughput, i.e. using the access network connection with higher throughput, or multiple access network connection simultaneously. 3 = redundancy, i.e. sending duplicated (redundancy) packets over multiple access network connections for high-reliability and low-latency applications. 4 = QoS, i.e. performing MTS based on the QoS requirement (qoS).
trafficSplit	Structure	0..N	Indicates the requested ratio between multiple access networks for the network aggregation required for higher throughput. This is based on the existing 3GPP ATSSS feature specified in clause 5.32.8 of ETSI TS 123 501 [i.5].
>accessId	Unit32	1	Unique identifier for the access network connection.
>splittingRatio	String	1	Splitting ratio for the access network as per the accessId.

Element	Type	Cardinality	Description
trafficSwitch	Structure	0..1	Priority based access network connection switching for seamless handover as per the application preference. This is based on the existing 3GPP ATSSS feature specified in clause 5.32.8 of ETSI TS 123 501 [i.5].
>accessId	Uint32	1	Unique identifier for the access network connection.
trafficDirection	String	1	The direction of the requested MTS session: 00 = Downlink (towards the UE). 01 = Uplink (towards the application/session). 10 = Symmetrical (see note).
NOTE:	For the downlink direction of a symmetrical flow, "sourceIp" and "sourcePort" in the "flowFilter" structure are used for source address and port, respectively; "dstIp" and "dstPort" are used for destination address and port, respectively. For the uplink direction of a symmetrical flow, "sourceIp" and "sourcePort" are used for destination address and port, respectively; "dstIp" and "dstPort" are used for source address and port, respectively.		

7.3 Subscription data types

7.3.1 Introduction

This clause defines data structures to be used for subscription.

7.3.2 Type: BwChgEventSubscription

This type represents a subscription to the notifications from BWM service about the bandwidth utility or data volume dispersion information.

The attributes of the BwChgSubscription shall follow the indications provided in table 7.3.2-1.

Table 7.3.2-1: Attributes of the BwChgEventSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "BwChgEventSubscription".
callbackReference	Uri	0..1	URI exposed by the client on which to receive notifications via HTTP. See note 1.
websocketNotifConfig	WebsocketNotifConfig	0..1	Provides details to negotiate and signal the use of a WebSocket connection between BWM 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 BWM API as it acts as an ID for the subscription.
filterCriteria	Structure (inlined)	1..N	List of filtering criteria for the BW change event subscription. Any filtering criteria from below, which is included in the request, shall also be included in the response.
>applnId	String	1	Application instance identifier.
>uelp	String	0..1	Source address identity of session. See note 2.
NOTE 1:	At least one of callbackReference and websocketNotifConfig shall be provided by the service consumer. If both are provided, it is up to BWM service to choose an alternative and return only that alternative in the response, as described in ETSI GS MEC 009 [6], clause 6.12a.		
NOTE 2:	For the case of session specific bw allocation, uelp shall be used to identify subscription per UE per application.		

7.3.3 Type: SubscriptionLinkList

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

The attributes of the SubscriptionLinkList shall follow the indications provided in table 7.3.3-1.

Table 7.3.3-1: Attributes of the SubscriptionLinkList

Attribute name	Data type	Cardinality	Description
_links	Structure (inlined)	0..1	List of hyperlinks related to the resource.
>self	LinkType	1	URI of this resource.
>subscriptions	Structure (inlined)	0..N	The service consumer's subscriptions.
>>href	Uri	1	The URI referring to the subscription.
>>subscriptionType	String	1	Type of the subscription. The string shall be set according to the "subscriptionType" attribute of the associated subscription data type event defined in clause 7.3.

7.4 Notification data types

7.4.1 Introduction

This clause defines data structures to be used for notification.

7.4.2 Type: BwChgEventNotification

This type represents a subscription to the notifications from BWM service about the bandwidth utility or the data volume across applications to obtain per UE information. The NWDAF in ETSI TS 123 288 [i.4] provides data volume dispersion information that shows the percentage of data traffic volume that a UE, or a group of UEs, or any UE, generated at a location or in a slice during the period of interest.

The attributes of the BwChgEventNotification shall follow the indications provided in table 7.4.2-1.

Table 7.4.2-1: Attributes of the BwChgEventNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "BwChgEventNotification".
timeStamp	Structure (inlined)	0..1	Time stamp to indicate when the corresponding information elements are sent.
>seconds	Uint32	1	The seconds part of the Time. Time is defined as Unix-time since January 1, 1970, 00:00:00 UTC.
>nanoSeconds	Uint32	1	The nanoseconds part of the Time. Time is defined as Unix-time since January 1, 1970, 00:00:00 UTC.
dataVolDispersion	Structure (inlined)	1..N	List of data volume dispersion information.
>dataVol	Uint32	1	Sum of data volume (UL/DL) exchanged per UE per applications, ETSI TS 123 288 [i.4]. See note.
>appInstId	String	1	Application instance identifier.
>uelp	String	0..1	Source address identity of session.
_links	Structure (inlined)	0..1	Link to resource related to this notification.
>subscription	LinkType	1	A link to related subscription.
NOTE:	The dataVolDispersion can be reported either as the total data volume of the session or periodically. dataVolDispersion shall provide information on data consumption per UE, in case of higher/lower data consumption, the service consumer can update the allocated bandwidth.		

7.5 Referenced structure data types

7.5.1 Introduction

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

7.5.2 Type: LinkType

This type represents a type of link.

Table 7.5.2-1: Attributes of the LinkType

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

7.5.3 Type: WebsocketNotifConfig

This type represents configuration for the delivery of subscription notifications over Websockets per the pattern defined in clause 6.12a of ETSI GS MEC 009 [6].

Table 7.5.3-1: Attributes of the WebsocketNotifConfig

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

8 BWM API definition

8.1 Introduction

This clause defines the resources and operations of the Bandwidth Management API (BWM API).

8.2 Global definitions and resource structure

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

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

Where:

- The "apiRoot" and "apiName" consists of the scheme ("https"), host and optional port, and an optional prefix string. It can be discovered using the service registry.
- The "apiName" shall be set to "bwm".
- The "apiVersion" shall be set to "v1" for the present document. All resource URIs in the clauses below is defined relative to the above root URI.

The API shall support HTTP over TLS as defined in clause 6.22 of ETSI GS MEC 009 [6].

The content format of JSON shall be supported.

The JSON format is 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 [6]. 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 [6] for more information.

Figure 8.2-1 illustrates the resource URI structure of this API.

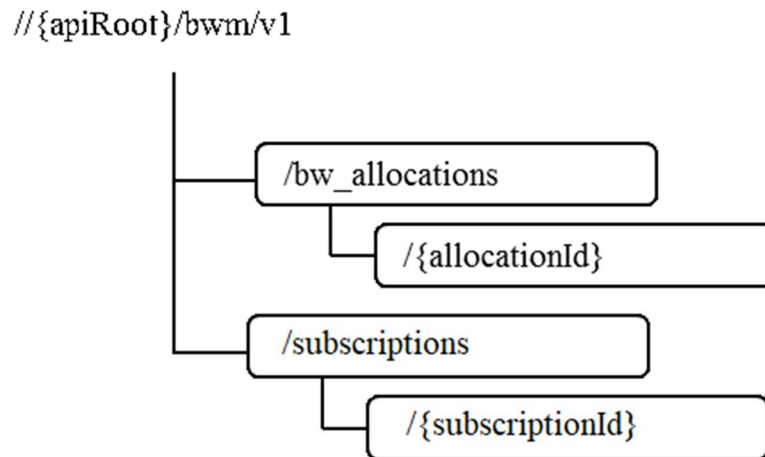


Figure 8.2-1: Resource URI structure of the BWM API

Table 8.2-1 provides an overview of the resources defined by the present document, and the applicable HTTP methods.

Table 8.2-1: HTTP methods overview

Resource name	Resource URI	HTTP method	Meaning
A list of bandwidthAllocation	/bw_allocations	GET	Retrieve information about a list of bandwidthAllocation resources.
		POST	Create a bandwidthAllocation resource.
Individual bandwidthAllocation	/bw_allocations/{allocationId}	GET	Retrieve information about a specific bandwidthAllocation.
		PUT	Update the information about a specific bandwidthAllocation.
		PATCH	Modify the information about a specific existing bandwidthAllocation by sending updates on the data structure.
		DELETE	Remove a specific bandwidthAllocation.
All subscriptions for a subscriber	/subscriptions	GET	Retrieve a list of active subscriptions for this subscriber.
		POST	Create a new subscription.
Existing subscription	/subscriptions/{subscriptionId}	GET	Retrieve information on current specific subscription.
		PUT	Modify existing subscription by sending a new structure.
		DELETE	Cancel the existing subscription.
Notification callback	Client provided callback reference	POST	Send a notification.

8.3 Resource: individual bandwidthAllocation

8.3.1 Description

This resource is used to represent a bandwidth allocation instance, which follows the resource data type of "BwInfo" as specified in clause 7.2.2.

8.3.2 Resource definition

Resource URI: {apiRoot}/bwm/v1/bw_allocations/{allocationId}

Resource URI Variables for this resource are defined in table 8.3.2-1.

Table 8.3.2-1: Resource URI Variables for resource "individual bandwidthAllocation"

Name	Definition
apiRoot	See clause 8.2.
allocationId	Represents a bandwidth allocation instance.

8.3.3 Resource Methods

8.3.3.1 GET

This method retrieves information about a bandwidthAllocation resource. This method is typically used in "Get configured bandwidth allocation from Bandwidth Management Service" procedure as described in clause 6.2.5.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 8.3.3.1-1 and 8.3.3.1-2.

Table 8.3.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 8.3.3.1-2: 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
	BwInfo	1	200 OK	It is used to indicate nonspecific success. The response body contains a representation of the 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	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.

8.3.3.2 PUT

This method updates the information about a bandwidthAllocation resource. As specified in ETSI GS MEC 009 [6], the PUT HTTP method has "replace" semantics.

PUT method is typically used in "Update requested bandwidth requirements on Bandwidth Management Service" procedure as described in clause 6.2.4.

PUT HTTP method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 8.3.3.2-1 and 8.3.3.2-2.

Table 8.3.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 8.3.3.2-2: Data structures supported by the PUT request/response on this resource

Request body	Data type	Cardinality	Remarks	
	BwInfo	1	BwInfo with updated information is included as entity body of the request.	
Response body	Data type	Cardinality	Response codes	Remarks
	BwInfo	1	200 OK	Upon success, a response body containing data type describing the updated BwInfo 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	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.
	ProblemDetails	1	406 Not Acceptable	It is used to indicate the server cannot provide any of the content formats supported by the clients. In the returned ProblemDetails structure, the "details" attribute should convey more information about the error.

8.3.3.3 PATCH

This method updates the information about a bandwidthAllocation resource. As specified in ETSI GS MEC 009 [6], the PATCH HTTP method updates a resource on top of the existing resource state by just including the changes ("deltas") in the request body.

PATCH method is used in "Update requested bandwidth requirements on Bandwidth Management Service" procedure.

PATCH HTTP method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 8.3.3.3-1 and 8.3.3.3-2.

Table 8.3.3.3-1: URI query parameters supported by the PATCH method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 8.3.3.3-2: Data structures supported by the PATCH request/response on this resource

Request body	Data type	Cardinality	Remarks	
	BwInfoDeltas	1	Description of the changes to instruct the server how to modify the resource representation.	
Response body	Data type	Cardinality	Response Codes	Remarks
	BwInfo	1	200 OK	Upon success, a response body containing data type describing the updated BwInfo 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, or the client is not authorized to perform it. More information shall be provided in the "details" 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.
	ProblemDetails	1	406 Not Acceptable	It is used to indicate the server cannot provide any of the content formats supported by the clients. In the returned ProblemDetails structure, the "details" attribute should convey more information about the error.

8.3.3.4 POST

Not supported.

8.3.3.5 DELETE

DELETE method is typically used in "Unregister from Bandwidth Management Service" procedure as described in clause 6.2.3.

DELETE HTTP method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 8.3.3.5-1 and 8.3.3.5-2.

Table 8.3.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 8.3.3.5-2: 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	
	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.

8.4 Resource: a list of bandwidthAllocations

8.4.1 Description

This resource is used to represent a list of mobile edge bandwidth allocations.

8.4.2 Resource definition

Resource URI: {apiRoot}/bwm/v1/bw_allocations

Resource URI Variables for this resource are defined in table 8.4.2-1.

Table 8.4.2-1: Resource URI Variables for resource a list of bandwidthAllocations

Name	Definition
apiRoot	See clause 8.2

8.4.3 Resource Methods

8.4.3.1 GET

This method retrieves information about a list of bandwidthAllocation resources. This method is typically used in "Get configured bandwidth allocation from Bandwidth Management Service" procedure as described in clause 6.2.5.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 8.4.3.1-1 and 8.4.3.1-2. When no URI query parameter is present, all the relevant bandwidthAllocations resources to the requestor will be returned.

Table 8.4.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
app_instance_id	String	0..N	A MEC application instance may use multiple app_instance_ids as an input parameter to query the bandwidth allocation of a list of MEC application instances. app_instance_id corresponds to appInstId defined in table 7.2.2-1. See note.
app_name	String	0..N	A MEC application instance may use multiple app_names as an input parameter to query the bandwidth allocation of a list of MEC application instances. app_name corresponds to appName defined in table 7.2.2-1. See note.
session_id	String	0..N	A MEC application instance may use session_id as an input parameter to query the bandwidth allocation of a list of sessions. session_id corresponds to allocationId defined in table 7.2.2-1. See note.

NOTE: Either "app_instance_id" or "app_name" or "session_id" or none of them shall be present.

Table 8.4.3.1-2: 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
	BwInfo	0..N	200 OK	Upon success, a response body containing an array of the bandwidthAllocations 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	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.

8.4.3.2 PUT

Not supported.

8.4.3.3 PATCH

Not supported.

8.4.3.4 POST

This method is used to create a bandwidthAllocation resource. This method is typically used in "Register to Bandwidth Management Service" procedure as described in clause 6.2.1.

POST HTTP method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 8.4.3.4-1 and 8.4.3.4-2.

Table 8.4.3.4-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 8.4.3.4-2: Data structures supported by the POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	BwInfo	1	Entity body in the request contains BwInfo to be created.	
Response body	Data type	Cardinality	Response codes	Remarks
	BwInfo	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.
	ProblemDetails	1	406 Not Acceptable	It is used to indicate the server cannot provide any of the content formats supported by the clients. In the returned ProblemDetails structure, the "details" attribute should convey more information about the error.

8.4.3.5 DELETE

Not supported.

8.5 Resource: subscriptions

8.5.1 Description

This resource contains various resources related to subscriptions for notifications.

8.5.2 Resource definition

Resource URI: {apiRoot}/bwm/v1/subscriptions

This resource shall support the resource URI variables defined in table 8.5.2-1.

Table 8.5.2-1: Resource URI variables for resource "subscriptions"

Name	Definition
apiRoot	See clause 8.2

8.5.3 Resource methods

8.5.3.1 GET

The GET method is used to request information about the subscriptions for this requestor. Upon success, the response contains an 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 8.5.3.1-1 and 8.5.3.1-2.

Table 8.5.3.1-1: 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: bw_chg: bw change information.

Table 8.5.3.1-2: 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
	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 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.

8.5.3.2 PUT

Not applicable.

8.5.3.3 PATCH

Not applicable.

8.5.3.4 POST

The POST method is used to create a new subscription to BWM notification. Upon success, the response contains an entity body describing the created subscription.

This method shall support the request and response data structures and response codes, as specified in table 8.5.3.4-1.

Table 8.5.3.4-1: Data structures supported by the POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	{NotificationSubscription}	1	Entity body in the request contains the data type of the specific BWM event subscription that is to be created, where the data type options are listed below and defined in clause 7.3: <ul style="list-style-type: none"> BwChgEventSubscription. 	
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 and as defined in clause 7.3: <ul style="list-style-type: none"> BwChgEventSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

	ProblemDetails	0..1	415 Unsupporte d 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 Unprocessa ble Entity	It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if a JSON request body is syntactically correct but semantically incorrect, for example if the 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.

8.5.3.5 DELETE

Not applicable.

8.6 Resource: existing subscription

8.6.1 Description

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

8.6.2 Resource definition

Resource URI: {apiRoot}/bwm/v1/subscriptions/{subscriptionId}

This resource shall support the resource URI variables defined in table 8.6.2-1.

Table 8.6.2-1: Resource URI variables for resource "existing subscription"

Name	Definition
apiRoot	See clause 8.2.
subscriptionId	Refers to created subscription, where the BWM API allocates a unique resource name for this subscription. The resource name can be also used to identify the resource.

8.6.3 Resource methods

8.6.3.1 GET

The GET method is used to retrieve information about this subscription. Upon success, the response contains an 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 8.6.3.1-1 and 8.6.3.1-2.

Table 8.6.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 8.6.3.1-2: 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
	{NotificationSubscription}	1	200 OK	Upon success, a response body containing the data type describing the specific BWM event subscription is returned. The allowed data types for subscriptions are defined in clause 7.3 and are as follows: <ul style="list-style-type: none"> BwChgEventSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

8.6.3.2 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 a 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 8.6.3.2-1 and 8.6.3.2-2.

Table 8.6.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 8.6.3.2-2: Data structures supported by the PUT request/response on this resource

Request body	Data type	Cardinality	Remarks	
	{NotificationSubscription}	1	New NotificationSubscription is included as entity body of the request. The allowed data types for subscriptions are defined in clause 7.3 and are as follows: <ul style="list-style-type: none"> BwChgEventSubscription. 	
Response body	Data type	Cardinality	Response Codes	Remarks
	{NotificationSubscription}	1	200 OK	Upon success, a response body containing the data type describing the updated subscription is returned. The allowed data types for subscriptions are defined in clause 7.3 and are as follows: <ul style="list-style-type: none"> BwChgEventSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	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 a 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.

8.6.3.3 PATCH

Not applicable.

8.6.3.4 POST

Not applicable.

8.6.3.5 DELETE

The DELETE method is used to cancel the existing subscription. Cancellation can be made by deleting the resource that represents the existing subscription.

This method shall support the URI query parameters, request and response data structures and response codes, as specified in tables 8.6.3.5-1 and 8.6.3.5-2.

Table 8.6.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 8.6.3.5-2: 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.

8.7 Resource: Notification callback

8.7.1 Description

This resource is used to represent a notification endpoint that the client has provided to receive BWM event notifications. The API producer can use this resource to send notifications related to BWM events to a subscribed API consumer.

8.7.2 Resource definition

The resource of callback URI is provided by the subscriber when subscribing to the notification.

Resource URI variables of this resource are defined in table 8.7.2-1.

Table 8.7.2-1: Resource URI variables for resource "Notification callback"

Name	Definition
n/a	

8.7.3 Resource methods

8.7.3.1 GET

Not applicable.

8.7.3.2 PUT

Not applicable.

8.7.3.3 PATCH

Not applicable.

8.7.3.4 POST

The POST method delivers a notification from the BWM service to the subscriber.

This method shall follow the provisions specified in tables 8.7.3.4-1 and 8.7.3.4-2 for URI parameters, request and response data structures, and response codes.

Table 8.7.3.4-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 8.7.3.4-2: Data structures supported by the POST request/response on this resource

	Data type	Cardinality	Remarks	
	{notification}	1	A notification of an event related to BWM information event The allowed data types are: <ul style="list-style-type: none"> BwChgEventNotification. 	
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a		204 No Content	The notification was delivered successfully. The response body shall be empty.
	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.	

8.7.3.5 DELETE

Not applicable.

9 MTS API definition

9.1 Introduction

This clause defines the resources and operations of the Multi-access Traffic Steering API (MTS API).

9.2 Global definitions and resource structure

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

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

Where:

- The "apiRoot" and "apiName" consists of the scheme ("https"), host and optional port, and an optional prefix string. It can be discovered using the service registry.
- The "apiName" shall be set to "mts".
- The "apiVersion" shall be set to "v1" for the present document. All resource URIs in the clauses below is defined relative to the above root URI.

The API shall support HTTP over TLS as defined in clause 6.22 of ETSI GS MEC 009 [6].

The content format of JSON shall be supported.

The JSON format is 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 [6]. 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 [6] for more information.

Figure 9.2-1 illustrates the resource URI structure of this API.

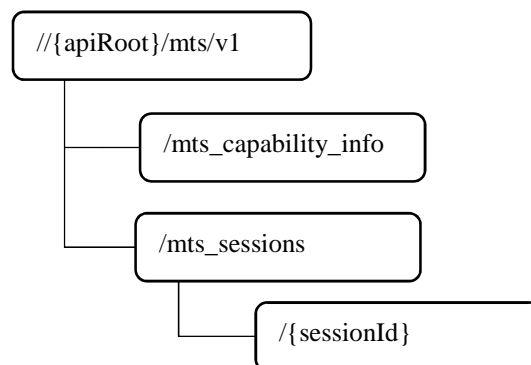


Figure 9.2-1: Resource URI structure of the MTS API

Table 9.2-1 provides an overview of the resources defined by the present document, and the applicable HTTP methods.

Table 9.2-1: HTTP methods overview

Resource name	Resource URI	HTTP method	Meaning
MTS capability information	/mts_capability_info	GET	Retrieve the MTS capability information.
Individual MTS session	/mts_sessions/{sessionId}	GET	Retrieve information about specific MTS session.
		PUT	Update the information about specific MTS session.
		DELETE	Remove specific MTS session.
A list of MTS sessions	/mts_sessions	GET	Retrieve information about a list of MTS sessions.
		POST	Create a MTS session.

9.3 Resource: MTS information

9.3.1 Description

This resource is used to represent a MTS service instance, which follows the resource data type of "MtsCapabilityInfo" as specified in clause 7.2.4.

9.3.2 Resource definition

Resource URI: {apiRoot}/mts/v1/mts_capability_info

This resource shall support the resource URI variables defined in table 9.3.2-1.

Table 9.3.2-1: Resource URI Variables for resource "MTS information"

Name	Definition
apiRoot	See clause 9.2.

9.3.3 Resource Methods

9.3.3.1 GET

The GET method is used to query information about the MTS information. This method is typically used in the "Get MTS service Info from the MTS Service" procedure as described in clause 6.2.6.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 9.3.3.1-1 and 9.3.3.1-2.

Table 9.3.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 9.3.3.1-2: 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
	MtsCapabilityInfo	1	200 OK	Upon success, a response body containing the MTS capability information is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed in the request. This error condition can also occur if the target area for the request is considered too large. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	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	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.

9.4 Resource: individual MTS session

9.4.1 Description

This resource is used to represent a MTS instance, which follows the resource data type of "MtsSessionInfo" as specified in clause 7.2.5.

9.4.2 Resource definition

Resource URI: {apiRoot}/mts/v1/mts_sessions/{sessionId}

This resource shall support the resource URI variables defined in table 9.4.2-1.

Table 9.4.2-1: Resource URI Variables for resource "individual MTS session"

Name	Definition
apiRoot	See clause 9.2.
sessionId	Represents a MTS session instance.

9.4.3 Resource Methods

9.4.3.1 GET

This method retrieves information about an individual MTS session. This method is typically used in the "Get configured MTS Session Info from the MTS Service" procedure as described in clause 6.2.10.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 9.4.3.1-1 and 9.4.3.1-2.

Table 9.4.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 9.4.3.1-2: 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
	MtsSessionInfo	1	200 OK	It is used to indicate nonspecific success. The response body contains a representation of the 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	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.

9.4.3.2 PUT

This method updates the information about an individual MTS session. As specified in ETSI GS MEC 009 [6], the PUT HTTP method has "replace" semantics.

PUT method is typically used in the "Update requested requirements on the MTS Service" procedure as described in clause 6.2.9.

PUT HTTP method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 9.4.3.2-1 and 9.4.3.2-2.

Table 9.4.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 9.4.3.2-2: Data structures supported by the PUT request/response on this resource

Request body	Data type	Cardinality	Remarks	
	MtsSessionInfo	1	MtsSessionInfo with updated information is included as entity body of the request.	
Response body	Data type	Cardinality	Response codes	Remarks
	MtsSessionInfo	1	200 OK	Upon success, a response body containing data type describing the updated MtsSessionInfo 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	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.	

9.4.3.3 DELETE

DELETE method is typically used in "Unregister from the MTS Service" procedure as described in clause 6.2.8.

DELETE HTTP method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 9.4.3.3-1 and 9.4.3.3-2.

Table 9.4.3.3-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 9.4.3.3-2: 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	
	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.	

9.5 Resource: a list of MTS sessions

9.5.1 Description

This resource is used to represent a list of MTS sessions.

9.5.2 Resource definition

Resource URI: {apiRoot}/mts/v1/mts_sessions

This resource shall support the resource URI variables defined in table 9.5.2-1.

Table 9.5.2-1: Resource URI Variables for resource "a list of MTS sessions"

Name	Definition
apiRoot	See clause 9.2.

9.5.3 Resource Methods

9.5.3.1 GET

This method retrieves information about a list of MTS sessions. This method is typically used in the "Get configured MTS Session Info from the MTS Service" procedure as described in clause 6.2.10.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 9.5.3.1-1 and 9.5.3.1-2.

Table 9.5.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
app_instance_id	String	0..N	A MEC application instance may use multiple app_instance_ids as an input parameter to query the MTS session of a list of MEC application instances. app_instance_id corresponds to appInstanceId defined in table 7.2.5-1. See note.
app_name	String	0..N	A MEC application instance may use multiple app_names as an input parameter to query the MTS session of a list of MEC application instances. app_name corresponds to appName defined in table 7.2.5-1. See note.
session_id	String	0..N	A MEC application instance may use session_id as an input parameter to query the information of a list of MTS sessions. session_id corresponds to sessionId defined in table 7.2.5-1. See note.
NOTE: Either "app_instance_id" or "app_name" or "session_id" or none of them shall be present.			

Table 9.5.3.1-2: 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
	MtsSessionInfo	0..N	200 OK	Upon success, a response body containing an array of the MTS sessions 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	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.

9.5.3.2 POST

This method is used to create a MTS session. This method is typically used in "Register application to the MTS Service" procedure as described in clause 6.2.7.

POST HTTP method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 9.5.3.2-1 and 9.5.3.2-2.

Table 9.5.3.2-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 9.5.3.2-2: Data structures supported by the POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	MtsSessionInfo	1	Entity body in the request contains MtsSessionInfo to be created.	
Response body	Data type	Cardinality	Response codes	Remarks
	MtsSessionInfo	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.
	ProblemDetails	1	405 Not Allowed	It is used when the client reached maximum sessions limit. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

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 Bandwidth Management API a supplementary description file compliant to the OpenAPI Specification [i.2].

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/gs015-bandwidth-mgmt-api>.

History

Document history		
V1.1.1	October 2017	Publication
V2.1.1	June 2020	Publication
V2.2.1	December 2022	Publication
V3.1.1	April 2024	Publication