ETSI GS MEC 015 V1.1.1 (2017-10)

GROUP SPECIFICATION

Mobile Edge Computing (MEC); Bandwidth Management API

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 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is 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 IPR Policy, no investigation, 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.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Mobile 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 Bandwidth Management mobile edge service. It describes the related application policy information including authorization and access control, information flows, required information and service aggregation patterns. 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]   ETSI GS MEC 001: "Mobile Edge Computing (MEC) Terminology ".

[2]   IETF RFC 2818: "HTTP Over TLS".

NOTE: Available at https://tools.ietf.org/html/rfc2818.

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

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

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

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

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

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

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

[7]   IETF RFC 7396: "JSON Merge Patch".

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: "Mobile Edge Computing (MEC); Technical Requirements".

[i.2]   OpenAPI Specification, Version 2.0, September 8, 2014.

NOTE: Available at https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md.

3   Definitions

For the purposes of the present document, the terms and definitions given in ETSI GS MEC 001 [1] apply.

4   Abbreviations

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

API	Application Programming Interface
BW	Bandwidth
BWM	Bandwidth Management
BWMS	Bandwidth Management Service
CDN	Content Delivery Network
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IETF	Internet Engineering Task Force
JSON	Javascript Object Notation
ME	Mobile Edge
OAI	Open API Initiative
RAN	Radio Access Network
REST	Representational State Transfer
RFC	Request For Comments
RTT	Round Trip Time
TLS	Transport Layer Security
URI	Uniform Resource Indicator
UTC	Coordinated Universal Time

5   Overview

The present document specifies the Bandwidth Management service (BWMS) API to support the requirements defined for Mobile Edge Computing in ETSI GS MEC 002 [i.1]. Clause 6 introduces how BWMS can be used by the mobile edge applications and by the mobile edge platform. It describes the information flows used for BWMS.

The information that can be exchanged over the Bandwidth Management (BWM) API is described in clause 7 which provides detailed description on all information elements that are used for BWMS.

Clause 8 describes the actual BWM API providing detailed information how information elements are mapped into a RESTful API design.

Figure 5-1 illustrates the mission of the BWMS, 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 requirements (bandwidth size, bandwidth priority, or both) for the whole application instance or different bandwidth requirements per session. The BWMS may aggregate all the requests and act in a manner that will help optimize the BW usage.
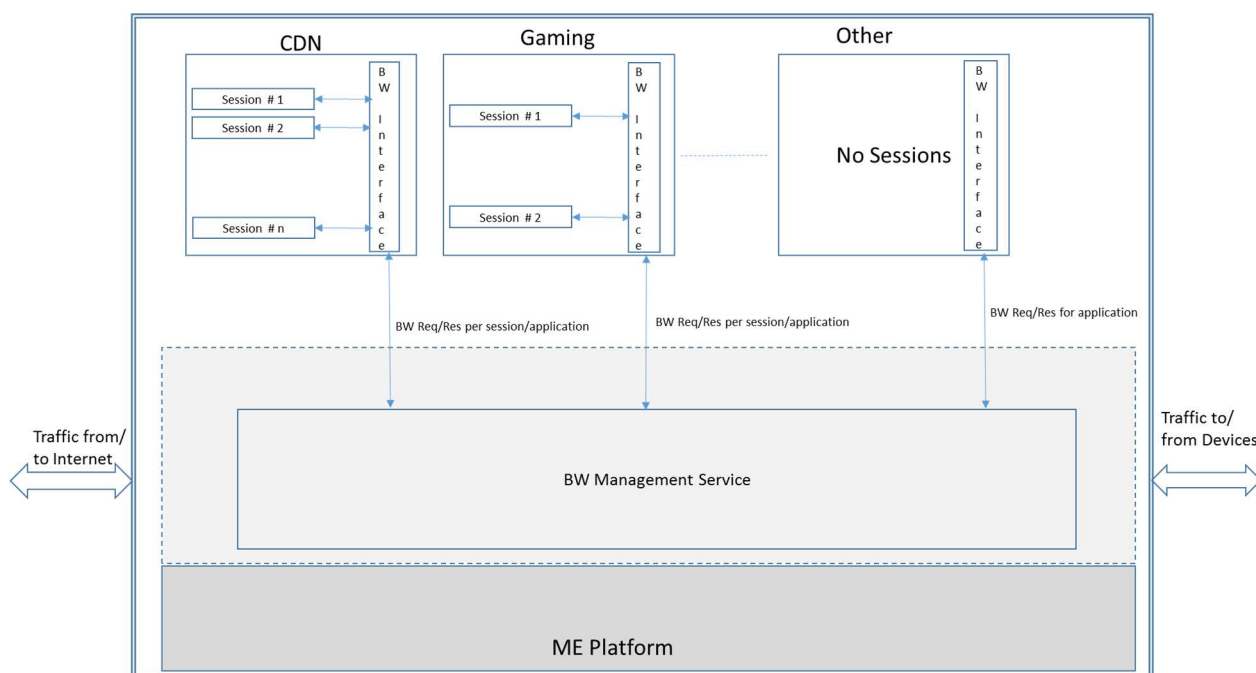
Figure 5-1: Bandwidth Management service description

6   Description of the service (informative)

6.1   Introduction

Different mobile edge applications running in parallel on the same mobile edge 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 session driven by application running from distant locations (outside the RAN). As all these applications and application sessions are competing over the same shared bandwidth resources, an optional central bandwidth resource allocator service may exist on the mobile edge platform.

The specific session or mobile edge 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 mobile edge applications can use BWMS to update/receive bandwidth information to/from the ME platform. The sequence diagrams that are relevant for BWMS are presented.

The BWM API enables all registered applications to statically and/or dynamically register for specific bandwidth allocation per session/application.

This specification of BWM API contains the HTTP protocol bindings for bandwidth 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 Mobile Edge Application registers to BWMS.
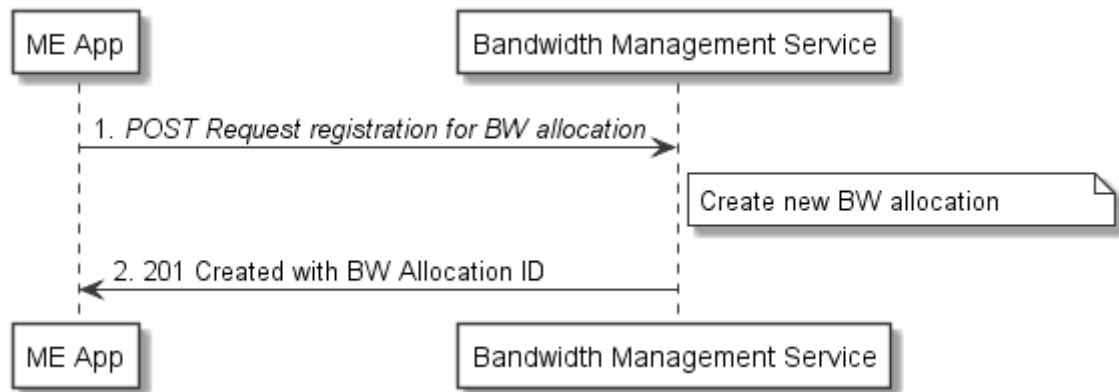
Figure 6.2.2-1: Flow of Mobile Edge Application registration to BWMS

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

1) Mobile edge 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 Mobile Edge Application Instance unregisters from BWMS.
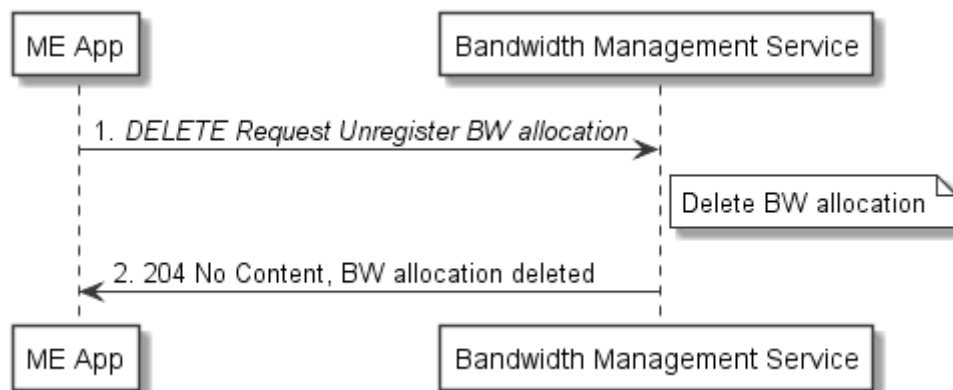
Figure 6.2.3-1: Flow of Mobile Edge Application unregistering BW allocation from BWMS

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

1) Mobile Edge Application Instance sends an unregister request to BWMS.

2) BWMS responds with an unregistration approval.

6.2.4 Update requested bandwidth requirements on Bandwidth Management Service

Figure 6.2.4-1 shows a scenario where a Mobile Edge Application Instance updates its requested bandwidth requirements on the BWMS.
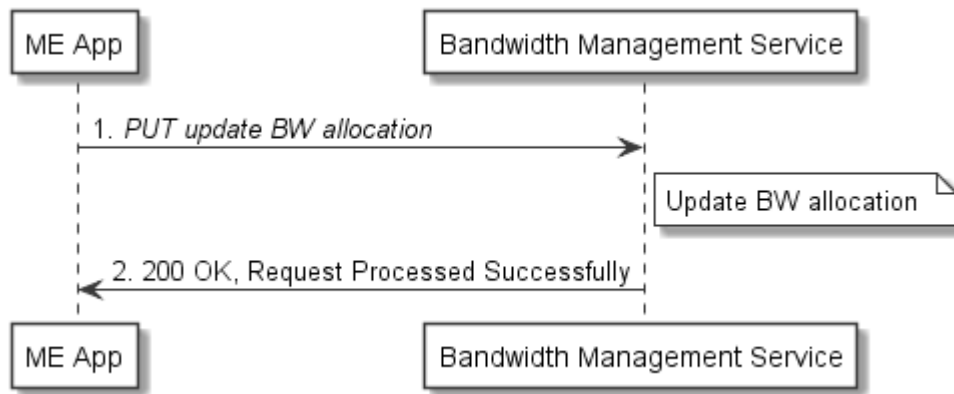
**Figure 6.2.4-1: Flow of Mobile Edge application updating its requested
bandwidth requirements on BWMS**

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

1) Mobile Edge 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 Bandwidth Management Service

Figure 6.2.5-1 shows a scenario where a Mobile Edge Application Instance gets its configured bandwidth allocation
from the BWMS.
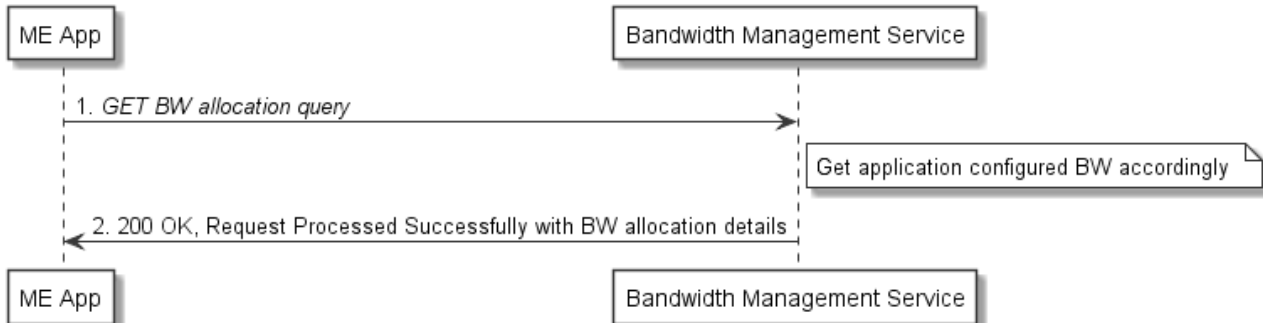
**Figure 6.2.5-1: Flow of Mobile Edge Application getting its configured
bandwidth allocation from BWMS**

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

1) Mobile Edge Application Instance sends a request to get its configured bandwidth allocation on the BWMS.

2) BWMS responds with the BW allocation 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
timeStamp	Structure (inlined)	0..1	Time stamp
>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
appInsId	String	1	Application instance identifier
requestType	Enum (inlined)	1	Numeric value (0 - 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
>sourceIp	String	0..1	Source address identity of session (including range)
>sourcePort	String	0..N	Source port identity of session
>dstAddress	String	0..1	Destination address identity of session (including range)
>dstPort	String	0..N	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

Conform 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
appInsId	String	1	Application instance identifier
requestType	Enum (inlined)	1	Numeric value (0 - 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
>sourceIp	String	0..1	Source address identity of session (including range)
>sourcePort	String	0..N	Source port identity of session
>dstAddress	String	0..1	Destination address identity of session (including range)
>dstPort	String	0..N	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

8   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}/bwm/v1/

The "apiRoot " is discovered using the service registry. The API shall support HTTP over TLS (also known as HTTPS [2]) (see IETF RFC 2818 [2]). TLS version 1.2 as defined by IETF RFC 5246 [3] shall be supported. HTTP is not recommended. All resource URIs in the sub-clauses below are defined relative to the above root URI.

The content format of JSON shall be supported.

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

This API shall require the use of the OAuth 2.0 client credentials grant type according to IETF RFC 6749 [4] with bearer tokens according to IETF RFC 6750 [5]. See clause 7.16 of ETSI GS MEC 009 [6] for more information. How the token endpoint and client credentials are provisioned into the mobile edge applications is out of scope of the present document.

This API supports additional application-related error information to be provided in the HTTP response when an error occurs. See clause 7.15 of ETSI GS MEC 009 [6] for more information.

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

//{apiRoot}/bwm/v1

/bw_allocations

/{allocationId}
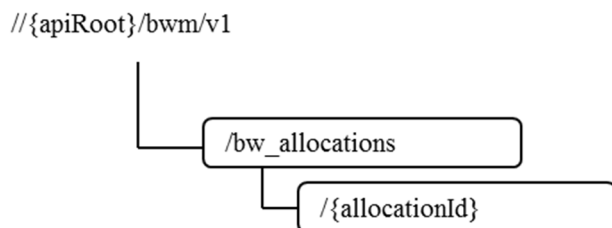
Figure 8.2-1: Resource URI structure of the BWM API

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

Table 8.2-1: HTTP methods overview

Resource name	Resource URI	HTTP method	Meaning
A list of bandwithAllocation	/bw_allocations	GET	Retrieve information about a list of bandwidthAllocation resources
		POST	Create a bandwidthAllocation resource
Individual bandwithAllocation	/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

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 the 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			
	Data type	**Cardinality**	**Response codes**	**Remarks**
Response body	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.

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 the 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.

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 the 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.

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 the 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 bandwithAllocations

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 the 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 mobile edge application instance may use multiple app_instance_ids as an input parameter to query the bandwidth allocation of a list of mobile edge application instances. See note.
app_name	String	0..N	A mobile edge application instance may use multiple ser_names as an input parameter to query the bandwidth allocation of a list of mobile edge application instances. See note.
session_id	String	0..N	A mobile edge application instance may use session_id as an input parameter to query the bandwidth allocation of a list of sessions. 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			
	Data type	**Cardinality**	**Response codes**	**Remarks**
Response body	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.

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 the 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.

8.4.3.5   DELETE

Not supported.

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/gitweb.cgi/MEC.GS_015.git.

History

Document history		
V1.1.1	October 2017	Publication