ETSI GS MEC 021 V2.1.1 (2020-01)

Multi-access Edge Computing (MEC); Application Mobility Service API

Disclaimer

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

Reference
DGS/MEC-0021AppMobility

Keywords
application, MEC, mobility

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 normative deliverables 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) 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 provides a specification for end-to-end MEC application mobility support in a multi-access edge system. The present document describes information flows, required information and operations. The present document also 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: "Multi-access Edge Computing (MEC); Terminology".

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

[3] ETSI GS MEC 003: "Multi-access Edge Computing (MEC); Framework and Reference Architecture".

[4] ETSI GS MEC 009: "Multi-access Edge Computing (MEC); General principles for MEC Service APIs".

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

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

[7] ETSI GS MEC 010-2: "Multi-access Edge Computing (MEC); MEC Management; Part 2: Application lifecycle, rules and requirements management".

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

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

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

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

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 016: "Multi-access Edge Computing (MEC); UE application interface".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

3.2 Symbols

Void.

3.3 Abbreviations

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

AMS	Application Mobility Service
API	Application Programming Interface
GTP	GPRS Tunnelling Protocol
HTTP	HyperText Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
JSON	JavaScript Object Notation
RFC	Request For Comments
RNIS	Radio Network Information Service
S-App	Source - Application instance
S-DP	Source - Data Plane
S-MEP	Source - MEC Platforms
S-MEPM	Source - MEC Platform Manager
T-App	Target - Application instance
T-DP	Target - Data Plane
T-MEP	Target - MEC Platforms
T-MEPM	Target - MEC Platform Manager
TEID	Tunnel End point IDentifier
URI	Uniform Resource Indicator
VIM	Virtualization Infrastructure Manager

4 Specification level requirements

4.1 Introduction

Application mobility is a unique feature of MEC system, which supports relocation of user context and/or application instance from one MEC host to another, or between a MEC host and a Cloud, especially when the MEC host is attached to mobile operator's networks. As a mobile device connected to a mobile network moves around within the network, it can result in the device connecting to the network entity associated to a different MEC host from the serving host. Consequently, there is necessity of relocating the application instance and/or user context associated to the device to a new MEC host to continue offering the best performance of service.

ETSI GS MEC 002 [2] describes some use cases related to application mobility or smart relocation, and associated requirements for MEC system to relocate the application instance and/or context to the "right" MEC host for optimizing the performance.

Application mobility may involve multiple MEC functional entities to relocate application instances and transfer user and application specific information within or between the MEC systems. Relocation decisions may be based on device mobility, customer profiles, application preferences and/or MEC infrastructure capability.

4.2 Functional requirements

Table 4.2-1 summarizes the functional requirements related to application mobility specified in ETSI GS MEC 002 [2].

Table 4.2-1: Functional requirements

Numbering		Functional requirement description
AppMobility01	[Mobility-01]	The MEC system shall be able to maintain connectivity between a UE and an application instance when the UE performs a handover to another cell associated with the same MEC host.
AppMobility02	[Mobility-02]	The MEC system shall be able to maintain connectivity between a UE and an application instance when the UE performs a handover to another cell not associated with the same MEC host.
AppMobility03	[Mobility-03]	The MEC platform may use available radio network information to optimize the mobility procedures required to support service continuity.
AppMobility04	[Mobility-04]	The MEC platform may use available core network information to optimize the mobility procedures required to support service continuity
AppMobility05	[Connectivity-02]	The MEC system shall support two instances of a MEC application running on different MEC hosts to communicate with each other.
AppMobility06	[Connectivity-03]	The MEC platform shall be able to allow an authorized MEC application to communicate with another MEC application located on another MEC host.
AppMobility07	[SmartReloc-03]	When the MEC system supports the feature SmartRelocation, the MEC management shall support the relocation of a MEC application instance from one MEC host to a different host within the system.
AppMobility08	[SmartReloc-04]	When the MEC system supports the feature SmartRelocation, a MEC host may support the relocation of a MEC application instance from a different host (within the system) to this particular host, and from this particular host to a different host (within the system).
AppMobility09	[SmartReloc-05]	When the MEC system supports the feature SmartRelocation, the system shall be able to move MEC application instances between MEC hosts in order to continue to satisfy the requirements of the MEC application.
AppMobility10	[SmartReloc-06]	When the MEC system supports the feature SmartRelocation, and based on a request from the UE, the system shall be able to relocate a MEC application running in a cloud environment to a MEC host fulfilling the requirements of the MEC application, and relocate a MEC application from a MEC host to a cloud environment outside the MEC system.
NOTE: The numbering of requirement in [ ] refers to the corresponding requirement in ETSI GS MEC 002 [2].		

5 Description of the services (informative)

5.1 Introduction

Application mobility service support may be considered as part of the service continuity support, for which the service to the user will resume and continue when the application instance is made available in the target MEC host and the user context, if needed, is transferred to the application instance there.

The characteristics of the service produced by the server application determines whether or not user context transfer is required for service continuity. For a stateless server application there is no state, i.e. user context, to transfer. For a stateful server application the user context may have to be transferred to the target application instance.

NOTE 1: The specification of the user context is outside the scope of the present document.

Application mobility support includes the following high level actions: the instantiation of the application in the target MEC host, if needed, and the transfer of user context, if needed, to the target application instance.

NOTE 2: The scenario of application mobility between two MEC systems and between the MEC system and an external cloud system is not specified in the present document.

Application mobility may involve multiple functional entities in MEC system, depending on different implementation approaches:

1) Application self-controlled user context transfer: The application itself, i.e. the server application instance (i.e. MEC application), or the client side application instance, or the centralized cloud instance, if available, may synchronize the user context in the target server application instance when necessary.

NOTE 3: For server application instances to resynchronize the user context the precondition is for MEC to enable the connectivity between the peer server application instances.

NOTE 4: The determination of the need for synchronization as well as the synchronization of the user context are application implementation dependent, and are outside the scope of the present document.

2) Device application assisted user context transfer: Device application initiates/triggers the application mobility and keeps the user context in the client during the relocation. The MEC system is the decision maker about the application mobility. Once the application is instantiated on the new MEC host, the application client will communicate with the server application instance directly to transfer and synchronize the user context.

NOTE 5: The user context transfer and synchronization are outside the scope of the present document.

3) MEC assisted user context transfer: MEC system triggers the application mobility. MEC system may facilitate the transfer of the user context to the target application instance.

Support of application mobility also depends on the application capability. An application instance may be dedicated to serve a single user; or it may serve multiple users simultaneously, such as multicast service to a group of users, or broadcast service to all the users associated to the MEC host.

Clause 5 provides descriptions of service for the three high level approaches described above. In addition, high level information flows for application mobility in different scenarios are provided. The high level information flows are then split into individual procedures to be defined in the present specification or in other MEC specifications. When possible, it is recommended to reuse the existing procedures, data models and APIs for application mobility.

5.2 End to end application mobility information flows

The high level application mobility service information flow for intra MEC system is shown in figure 5.2-1.

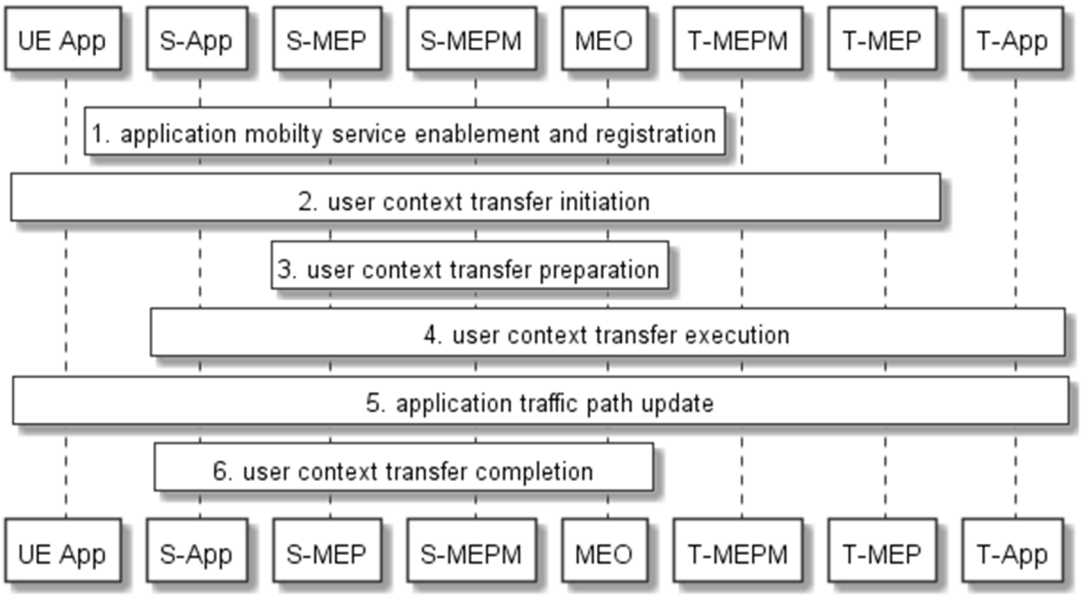


Figure 5.2-1: High level application mobility service information flow

The information flow of intra MEC system application mobility service may be divided into several sub-procedures that may or may not be present in the actual mobility scenario:

1) Application mobility enablement and registration: this sub-procedure illustrates the general procedure on enabling the application mobility service and allowing the application instances to register to the required application mobility services.

2) User context transfer initiation: this sub-procedure illustrates various detecting and triggering mechanisms for transferring the user context to the target application instance.

3) User context transfer preparation: this is an optional sub-procedure for MEC assisted user context transfer, and used for MEC system to prepare for the transfer.

4) User context transfer execution: this sub-procedure illustrates how the user context is transferred to and synchronized on the application instance running on the target MEC host.

5) Application traffic path update: this sub-procedure illustrates how MEC system reconfigures the data plane to redirect the traffic to the application instance on the target MEC host.

6) User context transfer completion: this sub-procedure illustrates how MEC system to clean-up the user context and/or application instance at source MEC host after the user context has been transferred.

The services like RNIS on the source MEC host and the target MEC host may be involved in the application mobility procedures. The detailed involvement will be described in the individual sub-procedures.

5.3 Application mobility enablement

The application mobility capability (e.g. UserContextTrasnferCapability) information may be included in the application descriptor (AppD) to indicate the stateful/stateless characteristic, the support of user context transfer, and the application mobility service dependency.

A suitable MEC host is selected based on the application requirements (including the application mobility support requirements) to instantiate the application. The application instance can register to the available AMS for application mobility support. The MEC system may also instantiate the same applications in other MEC host to assist the application mobility.

The information flow of application mobility service enablement and registration is shown in figure 5.3-1.

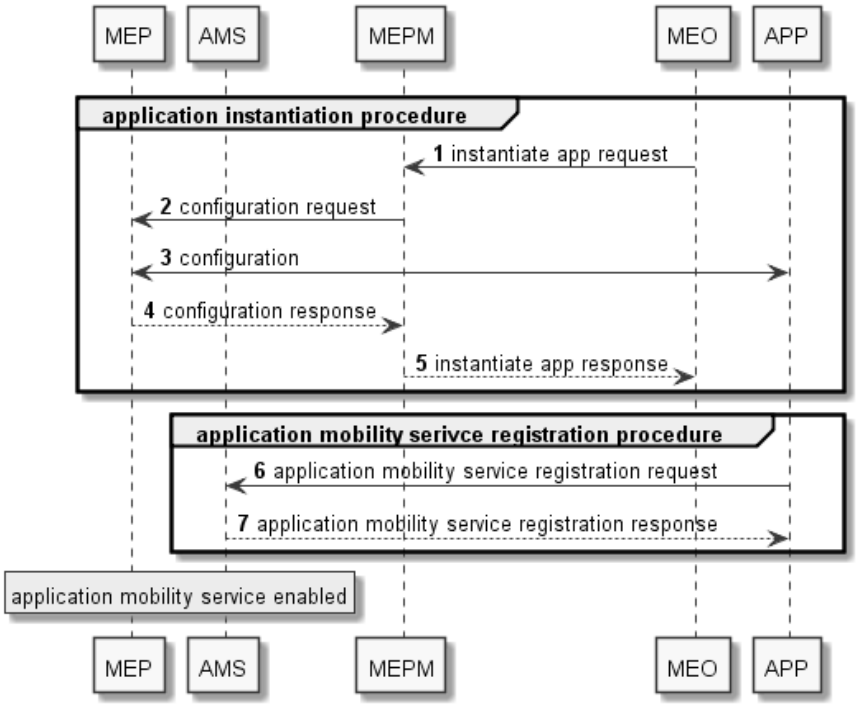


Figure 5.3-1: Application mobility service enablement and registration

The steps 1 - 5 are existing procedures specified in ETSI GS MEC 010-2 [7] and ETSI GS MEC 011 [5]:

6) The application instance sends the application mobility service registration request to the AMS running on the MEC host.

7) The AMS sends the application mobility service registration response to the application instance with the application mobility service ID to confirm the service registration success. The application mobility service is then enabled to serve to this application instance.

5.4 Application relocation initiation

5.4.1 Overview

Application mobility service support may rely on many factors, and may be initiated by different functional entities in the MEC system, including:

1) A combination of source and target MEPs and their associated services. Specific combinations include S-MEP & S-RNIS, S-MEP & S-DP, T-MEP & T-RNIS, T-MEP & T-DP, and the MEO.

2) A MEC application instance.

3) A UE application client.

A service of particular relevance to application mobility is RNIS which provides the services of radio network information to AMS. The information used to trigger application mobility services may include:

- information about UEs connected to the radio node(s) associated with the MEC host, and the related radio access bearers;
- changes in information related to UEs connected to the radio node(s) associated with the MEC host and the information related radio access bearers.

Using RNIS, the AMS is able to query for radio information or subscribe to notifications related to special events, a particular UE, or to radio node(s) attached to the MEC host.

RNIS uses a service consumer specified associateId to identify a particular UE or UE(s). The identifiers of the associateId by RNIS are:

- UE IPv4 address;
- UE IPv6 address;
- NATed IP address; or
- GTP TEID.

5.4.2 MEC assisted application mobility information flow

5.4.2.1 S-MEP triggered application mobility using RNIS

The first step in this flow is the AMS in the serving MEP (S-MEP) subscribing to cell change notifications for a UE or UEs in the cell(s) (radio nodes) associated to the MEC host. When a tracked UE moves across cells' boundary of the underlying network, the RNIS of serving MEC host (i.e. S-RNIS) will send event notifications about cell changes to the AMS in S-MEP. This may trigger application mobility procedures. Based on the received cell change notifications, the AMS in S-MEP verifies whether the UE has moved out of the coverage area of the source MEC host. If it does, the AMS in S-MEP will initiate application mobility procedures toward the T-MEH. The AMS in S-MEP uses the associateId in the notification to identify the target UE.

The S-MEP (i.e. AMS) initiated application mobility information flow regarding to UE cell change (handover) is depicted in figure 5.4.2.1-1.

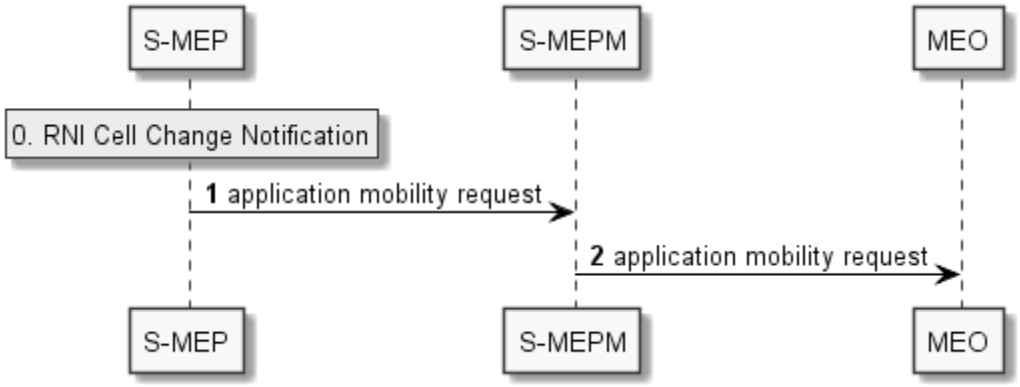


Figure 5.4.2.1-1: The information flow of S-MEP initiated application mobility

The information flow of S-MEP (i.e. AMS) initiated application mobility consists of following steps:

0) The AMS in S-MEP, registered by the application instance, subscribes the cell change notification associated with a UE or UEs in the cells under the MEC host. The AMS in S-MEP maps the appInstanceId with the associateId(s) after subscription. When a specified UE moves within the underlying network and triggers a cell change event, the S-RNIS sends a RNI cell change notification that indicates the handover status of the UE.

1) The associateId in the cell change notification can identify the UE that is performing the handover. The AMS in S-MEP processes the received cell change notification, mapping the notification to the application instance(s) serving the UE. The AMS in S-MEP may correlate different notifications to determine whether the UE has moved out of the coverage area of the S-MEH. If it does, the AMS in S-MEP sends to the MEO through the S-MEPM the MobilityProcedureNotification including the UE ID (associateId), the application instance IDs (appInstanceId), the source radio node ID (srcEcgi) and the target radio node ID (trgEcgi) which are all reported in the RNI cell change notification.

2) The S-MEPM relays the MobilityProcedureNotification to the MEO.

5.5 Application relocation verification and validation

When a UE moves to the service area of another MEC host, the MEC may instantiate on that MEC host the same application as the one serving to the UE, if an instance of the same application does not exist. The application relocation verification and validation are not addressed in the present document.

5.6 User context transfer

5.6.1 Introduction

For service continuity of a stateful application service, it is necessary to import the user context from the source application instance into the target application instance in the target MEC host. The user context includes user specific runtime data. The user context can be associated with a specific user or a group of users.

As specified in clause 5.1, there are three high level implementation approaches for user context transfer where the MEC system is the decision maker and selects appropriate MEC application instance:

1) Device application assisted state transfer

2) MEC assisted state transfer

3) Application self-controlled state transfer

The user context transfer is dependent on the capabilities of the application itself and of the underlying operating system of the MEC host. Both of these aspects are outside the scope of MEC specifications.

5.6.2 Application self-controlled user context transfer

The application self-controlled user context transfer assumes the application (server side, client side, centralized cloud component) to be able to detect the need for the user context transfer by its own means. Furthermore, it assumes the application is able to execute the context transfer without assistance from the MEC system. The role of the MEC system is to fulfil the applicable service and session continuity commitments for the application traffic and to enable the required application communication. The MEC application communication requirements and session and service continuity requirements may be brought into the MEC system at the time of on-boarding or deployment of the application.

An example scenario for the application self-controlled user context transfer is described below:

1) The MEC application instance serving the end user client is changed from one MEC host to another MEC host.

2) The session and service continuity requirement for the MEC application is to maintain the IP address.

3) The underlying access network and MEC reconfigure the routing of the application traffic so it reaches the new serving MEC application instance.

4) When the client connects with the server, the two have means to determine the need for the user context synchronization and have means to have the user context synchronized into the new serving MEC application instance, e.g. from the application client itself or from the central cloud instance.

5.6.3 Device application assisted user context transfer

The device application assisted user context transfer assumes that the application client is assisted by the device application associated with the corresponding MEC application (i.e. user application) in the MEC system.

NOTE: A device application is logically separate from the client application, irrespective of the way how the two have been implemented in the device. The device application session is with the user application LCM proxy which is a MEC system level functional entity.

The present Mx2 API [i.1] supports notifying the device application of the user application address change. Consequently the device application can receive the up to date information of the MEC application address and may pass this information to the client side application. A client application designed to be assisted by the device application does not require the underlying access network and MEC to maintain the IP address of the application. In addition, the client application may use the new MEC application instance address for the user context synchronization in the new user application instance.

5.6.4 MEC assisted user context transfer

The MEC assisted user context transfer relies on the application mobility service (AMS) of MEC to trigger the user context transfer and to inform the MEC application of the target end point of the user context.

The MEC application is a consumer of the AMS. The AMS is kept updated of the devices served by the MEC application. The AMS notifies the MEC application of the user context target end point when there is the need for a user context transfer. MEC application then sends the user context to the target end point. The user context is application specific and it is exchanged between MEC application peers in the source and target MEC hosts.

5.7 Application traffic path update

The application traffic path update is to switch the delivery of user traffic from the source to the target application instance. The AMS triggers the switch of traffic path after the application completes the user context transfer to the target application instance.

5.8 Application relocation completion

The application relocation completion is a process to clean-up the resource allocated to the application instance served to the application client in the source MEC host, if necessary.

The present document specifies the AMS API in clauses 6 and 7.

6 Sequence diagrams

6.1 Introduction

The service consumers, i.e. application instances or MEC platform, communicate with the application mobility service (AMS) over AMS API to register, deregister the application mobility service offered by the MEC system. The service consumers can update the user or other information of the registered application mobility service over the AMS API.

The service consumers may query the information of application mobility service in the MEC system, and subscribe to application mobility service on particular events in order to receive notifications of registered application mobility service when the events occur.

The application mobility service will provide the services to the registered consumers:

- endpoint information of adjacent application instances with communication links;
- identification of application instance running on the target MEC host;
- communication link information between the source and target instances of the same application;
- notification of application mobility status;
- assistance to clean up the user information at the source application instance and MEC platform when the user context has been transferred to the target application instance.

The AMS API supports the registration, queries and subscriptions over the RESTful API or over alternative transports such as message bus. Alternative transports are not specified in detail in the present document. When queries are used, the attribute-based filter expression may be used to limit the number of objects returned by query operation and attribute-selectors can be used to limit the number of attributes included in the response.

For RESTful architectural style, the present document defines the HTTP protocol bindings.

6.2 Register to application mobility service

The register to application mobility service allows a service consumer to register the application mobility service for one or more devices which share the same type of application mobility support.

Figure 6.2-1 shows a scenario where a service consumer (e.g. an MEC application or an MEC platform) registers to the application mobility service provided by the MEC system.

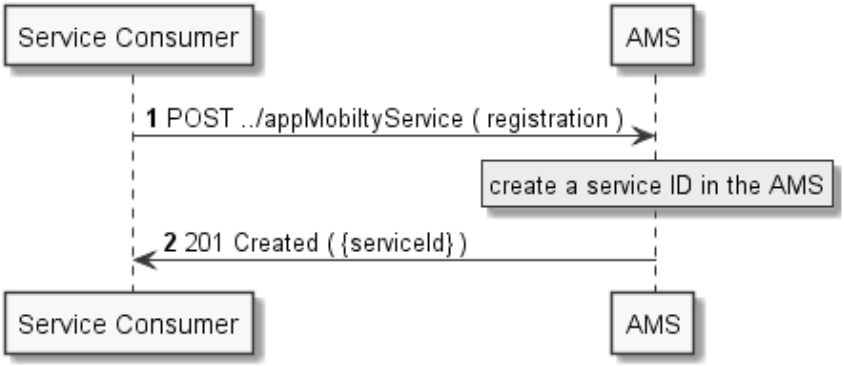


Figure 6.2-1: Flow of service consumer registration to AMS

A service consumer registration to AMS, as illustrated in figure 6.2-1, consists of the following steps:

1) A service consumer sends a POST request to register for the application mobility service provided by the MEC system.

2) The AMS creates a service ID of the registered application mobility service for the registered device(s). It then responds with the service ID for the approved registration of application mobility service.

6.3 Deregister to application mobility service

The deregister from application mobility service allows a service consumer to deregister from the application mobility service.

Figure 6.3-1 shows a scenario where a service consumer deregisters the application mobility service.

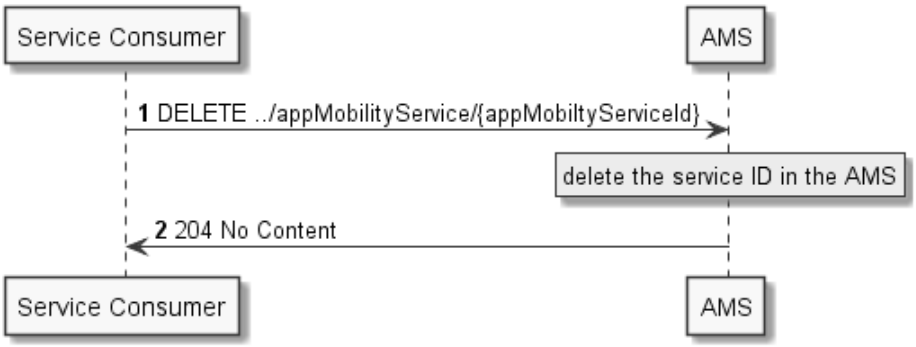


Figure 6.3-1: Flow of service consumer deregistration from AMS

A service consumer deregistration from AMS, as illustrated in figure 6.3-1, consists of the following steps:

1) A service consumer sends a request to deregister from the application mobility service for the service ID.

2) The AMS deletes the service ID and the information associated to the service consumer. It then responds with deregistration success.

6.4 Update application mobility service

The update application mobility service allows a service consumer to update information of the registered application mobility service for an individual or multiple devices.

Figure 6.4-1 shows a scenario where the service consumer (e.g. a MEC application or a MEC platform) update the registered application mobility service.

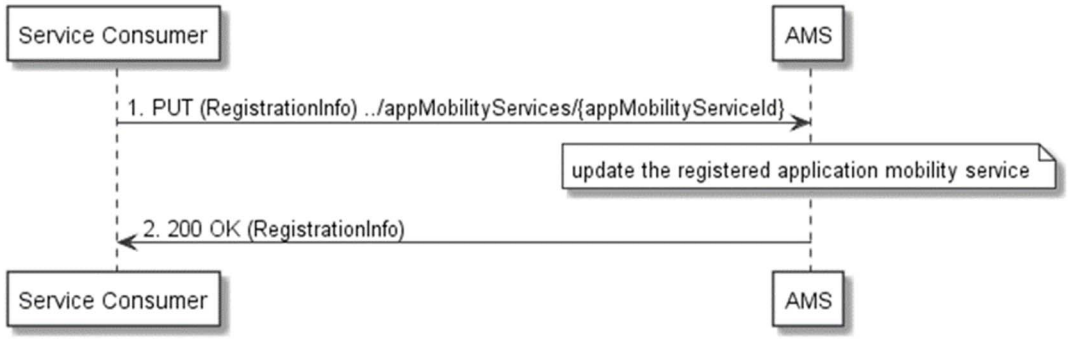


Figure 6.4-1: Flow of service consumer updating the registered application mobility service

A service consumer updating the registered application mobility service, as illustrated in figure 6.4-1, consists of the following steps:

1) A service consumer sends a request with RegistrationInfo to be updated for an associated service ID to update the registered application mobility service.

2) The AMS updates the information associated to the service ID and then responds with an update success with the updated RegistrationInfo.

6.5 User context transfer completion

The user context transfer completion allows the application instance to notify the AMS after the application completes the user context transfer from the source application instance to the target application instance. Therefore the AMS could start the next step to update the application mobility service.

Figure 6.5-1 shows a scenario that the application mobility service receives a user context transfer completion.

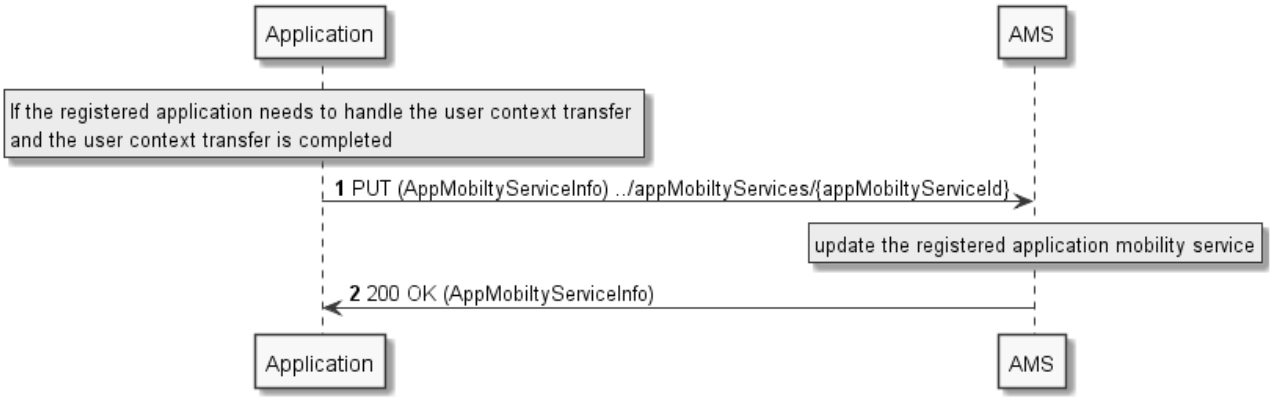


Figure 6.5-1: Flow of application instance sending the user context transfer completion notification

The user context transfer completion notification procedure, as illustrated in figure 6.5-1, consist of the following steps:

1) If the registered application instance needs to handle the user context transfer and the user context transfer is completed, the registered application instance sends a user context transfer completion to AMS.

2) The AMS updates the registered application mobility service for the device and sends a success message.

6.6 Subscribe to notifications of application mobility service

Subscribe to notifications of application mobility service allows a service consumer to receive notifications of application mobility service on particular events.

Figure 6.6-1 shows a scenario where the service consumer (e.g. a MEC application or a MEC platform) subscribes to notifications of application mobility service on particular events.

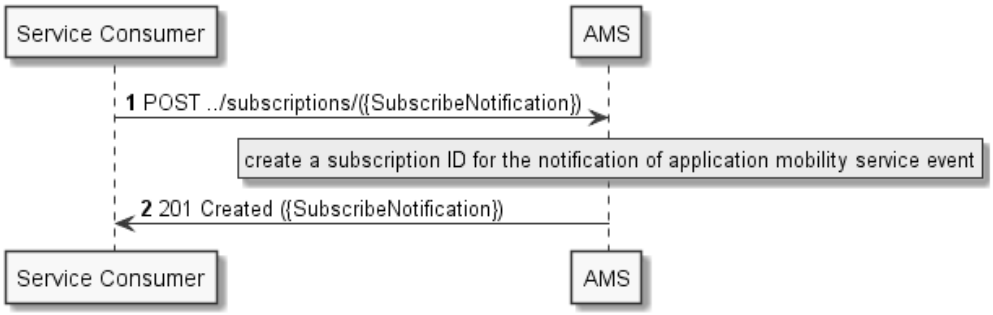


Figure 6.6-1: Flow of service consumer subscribing to notifications of application mobility service events

A service consumer subscribing to notifications of application mobility service events, as illustrated in figure 6.6-1, consists of the following steps:

1) A service consumer sends a POST request to subscribe to notifications of application mobility service events.

2) The AMS creates a subscription ID for the request and then responds with the subscription ID for the successful subscription.

6.7 Unsubscribe to notifications of application mobility service

The unsubscribe to notification of application mobility service allows a service consumer to unsubscribe to the notifications of application mobility service events.

Figure 6.7-1 shows a scenario where the service consumer unsubscribes to the notifications of application mobility service events.

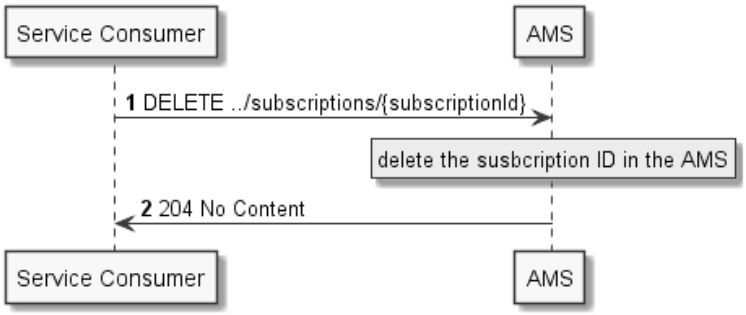


Figure 6.7-1: Flow of service consumer unsubscribing to the notifications of application mobility service

A service consumer unsubscribing to notification of application mobility service events, as illustrated in figure 6.7-1, consists of the following steps:

1) A service consumer sends a DELETE request to unsubscribe to the notifications of application mobility service events.

2) The AMS unsubscribes the notification of application mobility service events and deletes the subscription ID. It then responds with "204 No Content".

6.8 Update subscription to application mobility service notifications

Update subscription to application mobility service notification allows a service consumer to update subscription information to application mobility service notifications.

Figure 6.8-1 shows a scenario where the service consumer updates its subscription information to application mobility service notifications.

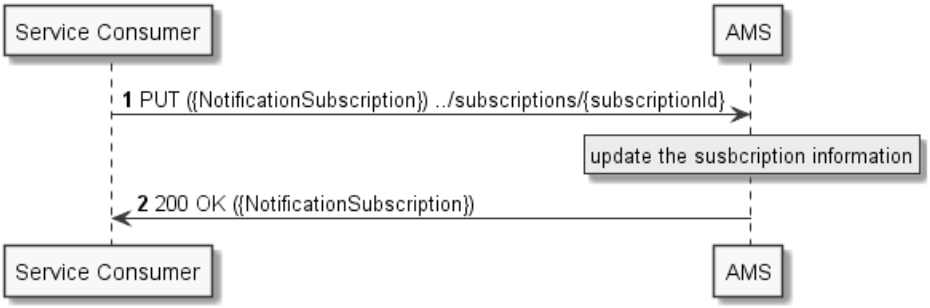


Figure 6.8-1: Flow of service consumer updating subscription to application mobility service notifications

A service consumer updating subscription information to application mobility service notifications, as illustrated in figure 6.8-1, consists of the following steps:

1) A service consumer sends a PUT request to update subscribe information to application mobility service notifications specified by the subscription ID.

2) The AMS updates the subscription information for application mobility service notifications, and then responds with "200 OK" to indicate the request processed successfully.

7 Data Models

7.1 Introduction

The following clauses describe the data models for support of application mobility service APIs.

7.2 Registration data types

7.2.1 Introduction

This clause defines data structures for application mobility service registrations.

7.2.2 Type: RegistrationInfo

This data type represents the Registration information used in the registration API of application mobility service.

Table 7.2.2-1: Attributes of RegistrationInfo

Attribute name	Data type	Cardinality	Description
appMobilityServiceId	String	0..1	The identifier of registered application mobility service. Shall be absent in POST requests, and present otherwise.
serviceConsumerId	Structure (inlined)	1	The identifier of service consumer requesting the application mobility service, i.e. either the application instance ID or the MEC platform ID.
>appInstanceId	String	0..1	If present, it represents the identifier of the application instance registering the application mobility service.
>mepId	String	0..1	If present, it represents the identifier of the MEC platform registering the application mobility service.
deviceInformation	Structure (inlined)	0..N	If present, it specifies the device served by the application instance which is registering the application mobility service.
>associateId	AssociateId	1	Represents the identifier of the device. See note.
>appMobilityServiceLevel	AppMobility ServiceLevel	0..1	If the application is stateful, this attribute shall be included.
>contextTransferState	Enum (inlined)	0..1	If present, it represents the state of transferring the user context to another application instance. The applicable values of this attribute are: 0 = NOT_TRANSFERRED (default value) 1 = USER_CONTEXT_TRANSFER_COMPLETED In the initial application mobility registration, this attribute shall be set to the default value.
expiryTime	Uint32	0..1	If present, it indicates the time of application mobility service expiration from the time of registration accepted. The value "0" means infinite time, i.e. no expiration. The unit of expiry time is one second.
NOTE: The data type of AssociateId is defined in ETSI GS MEC 012 [6].			

7.2.3 Type: AdjacentAppInstanceInfo

This data type represents the adjacent application instance information of the adjacentAppInstances resource.

Table 7.2.3-1: Attributes of AdjacentAppInstanceInfo

Attribute name	Data type	Cardinality	Description
appInstanceId	String	1	Identifier of the application instance.
appDId	String	1	Identifier of the application descriptor.
appInstanceCommLink	CommunicationInterface	1..N	It specifies the communication interface of application instance.
mecHostInformation	MECHostInformation	0..1	The MEC host where the application instance is running on.

7.3 Subscription data types

7.3.1 Introduction

This clause defines data structures for subscriptions.

7.3.2 Type: MobilityProcedureSubscription

This type represents a subscription to mobility procedure notifications from Application Mobility Service.

Table 7.3.2-1: Attributes of the MobilityProcedureSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "MobilityProcedureSubscription".
callbackReference	URI	1	URI selected by the service consumer to receive notifications on the subscribed Application Mobility Service. This shall be included both in the request and in response.
_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 AMS API as it acts as an ID for the subscription.
filterCriteria	Structure (inlined)	1	List of filtering criteria for the subscription. Any filtering criteria from below, which is included in the request, shall also be included in the response.
>appInstanceId	String	0..1	Identifier of the application instance that registers the application mobility service.
>associateId	AssociateId	0..N	0 to N identifiers to associate the information for specific UE(s) and flow(s).
>mobilityStatus	Enum (inlined)	0..N	In case mobilityStatus is not included in the subscription request, the default value 1 = INTER_HOST_MOBILITY_TRIGGERED shall be used and included in the response. 1 = INTERHOST_MOVEOUT_TRIGGERED. 2 = INTERHOST_MOVEOUT_COMPLETED. 3 = INTERHOST_MOVEOUT_FAILED.
expiryDeadline	TimeStamp	0..1	Time stamp.

7.3.3 Type: AdjacentAppInfoSubscription

This type represents a subscription to adjacent corresponding application instance information notifications from Application Mobility Service.

Table 7.3.3-1: Attributes of the AdjacentAppInfoSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "AdjacentAppInfoSubscription".
callbackReference	URI	1	URI selected by the service consumer to receive notifications on the subscribed Application Mobility Service. This shall be included both in the request and in response.
_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 AMS API as it acts as an ID for the subscription.
filterCriteria	Structure (inlined)	1	List of filtering criteria for the subscription. Any filtering criteria from below, which is included in the request, shall also be included in the response.
>appInstanceId	String	0..1	Identifier of the application instance that registers the application mobility service.
expiryDeadline	TimeStamp	0..1	Time stamp.

7.3.4 Type: SubscriptionLinkList

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

Table 7.3.4-1: Attributes of the SubscriptionLinkList

Attribute name	Data type	Cardinality	Description
_links	Structure (inlined)	1	List of hyperlinks related to the resource.
>self	LinkType	1	
subscription	Structure (inlined)	0..N	A link to a subscription.
>href	URI	1	The URI referring to the subscription.
>subscriptionType	Enum (inlined)	1	Numeric value (0 - 255) corresponding to specified type of subscription as following: 0 = RESERVED. 1 = MOBILITY_PROCEDURE. 2 = ADJACENT_APPINFO.

7.4 Notification data types

7.4.1 Introduction

This clause defines data structures that define notifications of application mobility service.

7.4.2 Type: MobilityProcedureNotification

This type represents a notification from AMS with regards to mobility procedure. The notification is sent by the Application Mobility Service to inform about the information of mobility procedure related to a UE.

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

Table 7.4.2-1: Attributes of the MobilityProcedureNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "MobilityProcedureNotification".
timeStamp	TimeStamp	0..1	Time stamp.
associateId	AssociateId	0..N	0 to N identifiers to associate the information for specific UE(s) and flow(s).
mobilityStatus	Enum (inlined)	1	Indicate the status of the UE mobility. Values are defined as following: 1 = INTERHOST_MOVEOUT_TRIGGERED. 2 = INTERHOST_MOVEOUT_COMPLETED. 3 = INTERHOST_MOVEOUT_FAILED. Other values are reserved.
targetAppInfo	Structure (inlined)	0..1	Identifiers to associate the information of target application instance.
>appInstanceId	String	0..1	Identifiers of the target application instance.
>commInterface	CommunicationInterface	0..1	If present, it specifies the communication interface of the application instance.

7.4.3 Type: AdjacentAppInfoNotification

This type represents a notification from AMS with regards to adjacent application instances information. The notification is sent by the Application Mobility Service to inform about the information of corresponding application instances in the adjacent MEC hosts.

The attributes of the AdjacentAppInfoNotification shall follow the indications provided in table 7.4.3-1.

Table 7.4.3-1: Attributes of the AdjacentAppInfoNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "AdjacentAppInfoNotification".
timeStamp	TimeStamp	0..1	Time stamp.
adjacentAppInfo	Structure (inlined)	0..N	0 to N identifiers to associate the information for adjacent application instances.
>appInstanceId	String	1	Identifiers of the target application instance.
>commInterface	CommunicationInterface	0..1	If present, it specifies the communication interface information of the application instance.

7.4.4 Type: ExpiryNotification

This data type represents a notification from AMS with regards to expiry of the existing subscription. The notification is sent by the Application Mobility Service regarding the information about expiry of a subscription.

Table 7.4.4-1: Attributes of the ExpiryNotification

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

7.4.5 Type: AppMobilityServiceLevel

This data type represents the application mobility service level information used in the application mobility service APIs.

Table 7.4.5-1: Attributes of AppMobilityServiceLevel

Attribute name	Data type	Cardinality	Description
appMobilityService Level	Enum (inlined)	1	This attribute provides an option for the application instance (server) to communicate with the application client before relocating this application instance to another MEC host. The applicable values of this attribute are: 1 = APP_MOBILITY_NOT_ALLOWED 2 = APP_MOBILITY_WITH_CONFIRMATION 3 = APP_MOBILITY_WITHOUT_CONFIRMATION. Other values are reserved. See note.

NOTE:

- APP_MOBILITY_NOT_ALLOWED: the current serving application instance continues to deliver the service to the device. The AMS will inform the MEP to perform the traffic steering to the device when the UE moves cross MEC hosts.
- APP_MOBILITY_WITH_CONFIRMATION: the serving application instance is allowed to change with the confirmation by the application once the device mobility happens. The AMS then will inform the MEP to update the traffic routing to the target application instance and clean up the previous resource allocated to the application instance serving to the device after the serving application instance has been changed.
- APP_MOBILITY_WITHOUT_CONFIRMATION: the serving application instance is allowed to change without confirmation from the application instance when the device mobility happens. The AMS will notify to the MEP to update the traffic routing to the target application instance and clean up the previous resource allocated to the application instance serving to the device after the serving application instance has been changed.

7.5 Referenced structured 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: CommunicationInterface

This type represents the communication interface of an application instance.

Table 7.5.2-1: Attributes of CommunicationInterface

Attribute name	Data type	Cardinality	Description
ipAddresses	Structure (inlined)	0..N	Entry point information of the service as one or more pairs of IP address and port.
>host	String	1	Host portion of the address.
>port	Int	1	Port portion of the address.

8 API Definition

8.1 Introduction

This clause defines the RESTFul resources and APIs for application mobility service.

8.2 Global definitions and resource structure

All resource URIs of APIs shall have the following root:

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

where:

- The "apiRoot" 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 "amsi" for application mobility service interface.
- The "apiVersion" shall be set to "v1" for the present document. All resource URIs in the clauses below are defined relative to the above root URI.

This API shall require the use of the OAuth 2.0 client credentials grant type according to IETF RFC 6749 [8] with bearer tokens according to IETF RFC 6750 [9]. See clause 7.16 of ETSI GS MEC 009 [4] for more information. How the token endpoint and client credentials are provisioned into the MEC applications is out of scope of the present document.

This API supports additional application-related error information to be provided in the HTTP response when an error occurs. See clause 7.15 of ETSI GS MEC 009 [4] for more information.

Figure 8.2-1 illustrates the resource URI structure of application mobility service interfaces.

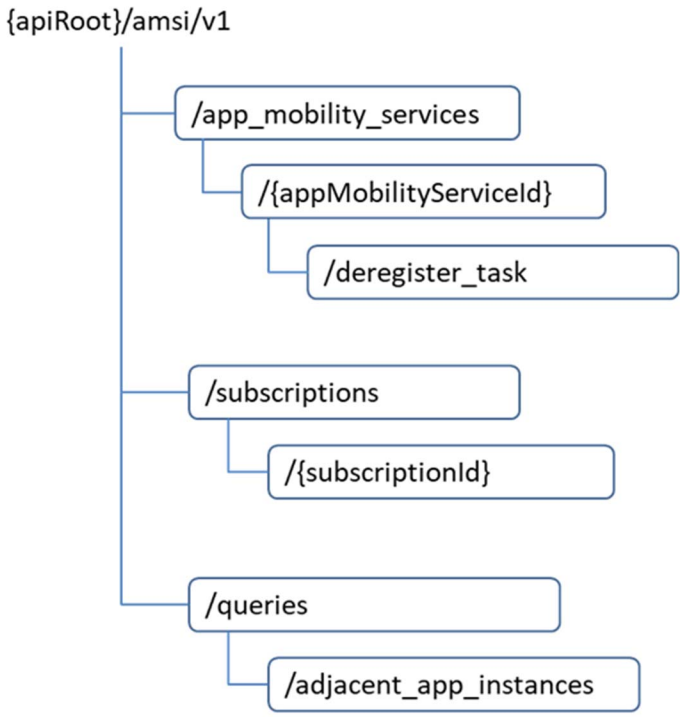


Figure 8.2-1: The resource URI structure of application mobility service interface

Table 8.2-1 summarizes the resources and associated HTTP methods for application mobility service APIs.

Table 8.2-1: Overview of resources and methods of application mobility service API

Resource name	Resource URI	HTTP Method	Description
Application mobility services	/app_mobility_services	POST	Create an application mobility service resource
		GET	Query multiple application mobility service resources
Individual application mobility service	/app_mobility_services/{appMobilityServiceId}	GET	Read an application mobility service resource
		DELETE	Delete an individual application mobility service resource
		PUT	Update an individual application mobility service resource
Deregister application mobility service task	/app_mobility_services/{appMobilityServiceId}/deregister_task	POST	Timer based individual deregistration of an application mobility service resource
Subscriptions See note	/subscriptions	POST	Subscribe to notifications of application mobility service event
		GET	Retrieve information of multiple subscriptions to notifications of application mobility service event
Individual subscription	/subscriptions/{subscriptionId}	GET	Read information of subscription of application mobility service
		DELETE	Terminate an individual subscription to notification of application mobility service event
		PUT	Update an individual subscription to notification of application mobility service event
Notification endpoint	(client provided)	POST	Notify about application mobility service event

Resource name	Resource URI	HTTP Method	Description
Adjacent application instances	/queries/adjacent_app_instances	GET	Read information of application instance(s) running on adjacent MEC host(s)
NOTE: Parent resources of all subscriptions related to the application mobility service.			

8.3 Resource: application mobility services

8.3.1 Description

This resource represents application mobility service. The service consumer can use the API to request resource allocation for registering the application mobility service and query the information of resource.

8.3.2 Resource definition

Resource URI: **{apiRoot}/amsi/v1/app_mobility_services**

This resource shall support the resource URI variables defined in table 8.3.2-1.

Table 8.3.2-1: Resource URI variables for resource

Name	Definition
apiRoot	See clause 8.2

8.3.3 Resource methods

8.3.3.1 GET

The GET method is to retrieve information about the registered application mobility service. Upon success, the response contains entity body with the list of application mobility services that are present for the requestor.

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

Name	Cardinality	Remarks
filter	0..1	Attribute-based filtering parameters according to ETSI GS MEC 011 [5]. The API producer shall support receiving filtering parameters as part of the URI query string. All attribute names that appear in the RegistrationInfo and in data types referenced from it shall be supported in attribute-based filtering parameters. See clause 6.19 in ETSI GS MEC 011 [5] for details.
all_fields	0..1	Include all complex attributes in the response. See clause 6.18 in ETSI GS MEC 011 [5] for details. The API producer shall support this parameter.
Fields	0..1	Complex attributes to be included into the response. See clause 6.18 in ETSI GS MEC 011 [5] for details. The API producer should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 6.18 in ETSI GS MEC 011 [5] for details. The API producer should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 6.18 in ETSI GS MEC 011 [5] for details. The API producer shall support this parameter. The following attributes shall be excluded from the RegistrationInfo structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: - 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	0	The method is to query the information of application instances.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	RegistrationInfo	0..N	200 OK	Information about zero or more application mobility services was queried successfully. The response body shall contain in an array the representations of zero or more application mobility services.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

8.3.3.2 PUT

Not applicable.

8.3.3.3 PATCH

Not applicable.

8.3.3.4 POST

The POST method is used to create a new application mobility service for the service requester. Upon success, the response contains entity body of created application mobility service with the service ID.

This method shall support the request and response data structures, and response codes, as specified in table 8.3.3.4-1.

Table 8.3.3.4-1: Data structures supported by the POST request/response on this resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	RegistrationInfo	1	The POST method is to create an application mobility service resource.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	RegistrationInfo	1	201 Created	An application mobility service with a service identifier was created successfully. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created application instance.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

8.3.3.5 DELETE

Not applicable.

8.4 Resource: individual application mobility service

8.4.1 Description

This resource represents individual application mobility service. A service consumer can use this API to query, update or delete the resource for a registered application mobility service.

8.4.2 Resource definition

Resource URI: **{apiRoot}/amsi/v1/app_mobility_services/{appMobilityServiceId}**

This resource shall support the resource URI variables defined in table 8.4.2-1.

Table 8.4.2-1: Resource URI variables for resource "individual app mobility service"

Name	Definition
apiRoot	See clause 8.2
appMobilityServiceId	It uniquely identifies the created individual application mobility service.

8.4.3 Resource methods

8.4.3.1 GET

The GET method is used to retrieve information about this individual application mobility service. Upon success, the response contains entity body with the data type describing the registered application mobility service.

This method shall support the URI input parameters, request and response data structures, and response codes, as specified in tables 8.4.3.1-1 and 8.4.3.1-2.

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

Name	Data type	Cardinality	Remarks
n/a			

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**
	RegistrationInfo	1	200 OK	Information about application mobility service was queried successfully.
	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.

	Data type	Cardinality	Response Codes	Remarks
Response body	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

8.4.3.2 PUT

The PUT method is used to update the existing individual application mobility service. PUT method in this case has "replace" semantics. Upon successful operation, the target resource is updated with new Data Type received within the message body of the PUT request.

NOTE 1: A service consumer, e.g. application instance, registers to the application mobility service. When its served clients change (such as adding a new user), the registered service consumer needs to update the registration information of the application mobility service.

NOTE 2: A registered stateful application instance updates its application mobility registration information (i.e. contextTransferState is set to USER_CONTEXT_TRANSFER_COMPLETED) after the user context of the application instance has been transferred to another application instance.

NOTE 3: A registered stateless application instance can update its application mobility registration information (i.e. contextTransferState is set to USER_CONTEXT_TRANSFER_COMPLETED) once the device is reported to move to the coverage of another MEC host and no user context transfer is required.

This method shall support the URI input parameters, request and response data structures, and response codes, as specified in tables 8.4.3.2-1 and 8.4.3.2-2.

Table 8.4.3.2-1: URI input parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 8.4.3.2-2: Data structures supported by the PUT request/response on this resource

	Data type	Cardinality	Response Codes	Remarks
Request body	RegistrationInfo	1		The updated information for the registered application mobility service resource.
Response body	RegistrationInfo	1	200 OK	Upon success, a response body containing data type describing the updated application mobility service.
	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.

	Data type	Cardinality	Response Codes	Remarks
Response body	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	412 Precondition Failed	It is used when a condition has failed during conditional requests, e.g. when using ETags to avoid write conflicts when using PUT. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	422 Unprocessable Entity	It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if an JSON request body is syntactically correct but semantically incorrect, for example if the target area for the request is considered too large. This error condition can also occur if the capabilities required by the request are not supported. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

8.4.3.3 PATCH

Not applicable.

8.4.3.4 POST

Not applicable.

8.4.3.5 DELETE

The DELETE method is used to deregister the individual application mobility service, and delete the resource that represents the individual application mobility service.

This method shall support the URI input parameters, request and response data structures, and response codes, as specified in tables 8.4.3.5-1 and 8.4.3.5-2.

Table 8.4.3.5-1: URI input parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 8.4.3.5-2: Data structures supported by the DELETE request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	n/a		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.

8.5 Resource: deregister application mobility service task

8.5.1 Description

This resource represents the deregister task of application mobility service. When the service consumer registers the application mobility service with the expiryTime attribute present and its value greater than 0, this resource is invoked and performs application mobility service deregistration on expiry of the timer.

NOTE: The deregister application mobility service task is used to cleanup the registered application mobility service resource allocated to the service consumer once the timer expires.

8.5.2 Resource definition

Resource URI: **{apiRoot}/amsi/v1/app_mobility_mervices/{appMobilityServiceId}/deregister_task**

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 "deregister application mobility service task"

Name	Definition
apiRoot	See clause 8.2
appMobilityServiceId	It uniquely identifies the created individual application mobility service

8.5.3 Resource methods

8.5.3.1 GET

Not applicable.

8.5.3.2 PUT

Not applicable.

8.5.3.3 PATCH

Not applicable.

8.5.3.4 POST

The POST method is to trigger deregistration of the application mobility service on expiry of the timer associated with the service.

This method shall support the URI input parameters, request and response data structures, and response codes, as specified in tables 8.5.3.4-1 and 8.5.3.4-2.

Table 8.5.3.4-1: URI input parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

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

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	n/a	0	204 No Content	The request is acknowledged. 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	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

8.5.3.5 DELETE

Not applicable.

8.6 Resource: subscriptions

8.6.1 Description

This resource contains various resources related to subscriptions for notifications.

8.6.2 Resource definition

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

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 "subscriptions"

Name	Definition
apiRoot	See clause 8.2

8.6.3 Resource methods

8.6.3.1 GET

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

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 8.6.3.1-1 and 8.6.3.1-2.

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

Name	Data type	Cardinality	Remarks
subscriptionType	String	0..1	Query parameter to filter on a specific subscription type. Permitted values: • mobility_proc Mobility Procedure • adj_app_info Adjacent Application Info

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

8.6.3.2 PUT

Not applicable.

8.6.3.3 PATCH

Not applicable.

8.6.3.4 POST

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

This method shall support the request and response data structures, and response codes, as specified in table 8.6.3.4-1.

Table 8.6.3.4-1: Data structures supported by the POST request/response on this resource

	Data type	Cardinality	Remarks	
Request body	{NotificationSubscription}	1	The entity body in the request contains data type of the specific AMS event subscription that is to be created, where the data type options are listed below and defined in clauses 7.3.2 through 7.3.4: • MobilityProcedureSubscription. • AdjacentAppInfoSubscription.	
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 clauses 7.3.2 through 7.3.4: • MobilityProcedureSubscription. • AdjacentAppInfoSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	415 Unsupported Media Type	It is used to indicate that the server or the client does not support the content type of the entity body. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

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

8.6.3.5 DELETE

Not applicable.

8.7 Resource: individual subscription

8.7.1 Description

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

8.7.2 Resource definition

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

This resource shall support the resource URI variables defined in table 8.7.2-1.

Table 8.7.2-1: Resource URI variables for resource "individual subscription"

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

8.7.3 Resource methods

8.7.3.1 GET

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

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 8.7.3-1-1 and 8.7.3-1-2.

Table 8.7.3-1-1: URI input parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 8.7.3-1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	{NotificationSubscription}	1	200 OK	Upon success, a response body containing data type describing the specific RNI event subscription is returned. The allowed data types for subscriptions are defined in clauses 7.3.2 through 7.3.4 and are as follows: • MobilityProcedureSubscription. • AdjacentAppInfoSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

8.7.3.2 PUT

The PUT method is used to update the existing individual subscription. PUT method in this case has "replace" semantics. Upon successful operation, the target resource is updated with new Data Type received within the message body of the PUT request.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 8.7.3-2-1 and 8.7.3-2-2.

Table 8.7.3-2-1: URI input parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 8.7.3-2-2: Data structures supported by the PUT request/response on this resource

	Data type	Cardinality	Response Codes	Remarks
Request body	{NotificationSubscription}	1		New NotificationSubscription is included as entity body of the request. The allowed data types for subscriptions are defined in clauses 7.3.2 through 7.3.4 and are as follows: • MobilityProcedureSubscription. • AdjacentAppInfoSubscription.
Response body	{NotificationSubscription}	1	200 OK	Upon success, a response body containing data type describing the updated subscription is returned. The allowed data types for subscriptions are defined in clauses 7.3.2 through 7.3.4 and are as follows: • MobilityProcedureSubscription. • AdjacentAppInfoSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	412 Precondition Failed	It is used when a condition has failed during conditional requests, e.g. when using ETags to avoid write conflicts when using PUT. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

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

8.7.3.3 PATCH

Not applicable.

8.7.3.4 POST

Not applicable.

8.7.3.5 DELETE

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

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 8.7.3-5-1 and 8.7.3-5-2.

Table 8.7.3-5-1: URI input parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 8.7.3-5-2: 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.

8.8 Resource: adjacent application instances

8.8.1 Description

This resource represents the information of application instances running on adjacent MEC host(s). A service consumer can use the API of the application mobility service to query the information about the application instance(s) on adjacent MEC host(s).

8.8.2 Resource definition

Resource URI: **{apiRoot}/amsi/v1/queries/adjacent_app_instances**

This resource shall support the resource URI variables defined in table 8.8.2-1.

Table 8.8.2-1: Resource URI variables for resource

Name	Definition
apiRoot	See clause 8.2

8.8.3 Resource methods

8.8.3.1 GET

The GET method is to retrieve the information of application instances running on adjacent MEC host(s). Upon success, the response contains entity body with the list of available application instance information.

This method shall support the URI input parameters, request and response data structures, and response codes, as specified in tables 8.8.3.1-1 and 8.8.3.1-2.

Table 8.8.3.1-1: URI input parameters supported by the GET method on this resource

Name	Cardinality	Remarks
filter	0..1	Attribute-based filtering parameters according to ETSI GS MEC 011 [5]. The API producer shall support receiving filtering parameters as part of the URI query string. All attribute names that appear in the AdjacentAppInstanceInfo and in data types referenced from it shall be supported in attribute-based filtering parameters. See clause 6.19 in ETSI GS MEC 011 [5] for details.
all_fields	0..1	Include all complex attributes in the response. See clause 6.18 in ETSI GS MEC 011 [5] for details. The API producer shall support this parameter.
fields	0..1	Complex attributes to be included into the response. See clause 6.18 in ETSI GS MEC 011 [5] for details. The API producer should support this parameter.
exclude_fields	0..1	Complex attributes to be excluded from the response. See clause 6.18 in ETSI GS MEC 011 [5] for details. The API producer should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes from the response. See clause 6.18 in ETSI GS MEC 011 [5] for details. The API producer shall support this parameter. The following attributes shall be excluded from the AdjacentAppInstanceInfo structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: - n/a

Table 8.8.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a	0	The method is to query the information of application instances.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	AdjacentAppInstanceInfo	0..N	200 OK	Information about zero or more adjacent application instances was queried successfully. The response body shall contain in an array the representations of zero or more adjacent application instance information.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

8.8.3.2 PUT

Not applicable.

8.8.3.3 PATCH

Not applicable.

8.8.3.4 POST

Not applicable.

8.8.3.5 DELETE

Not applicable.

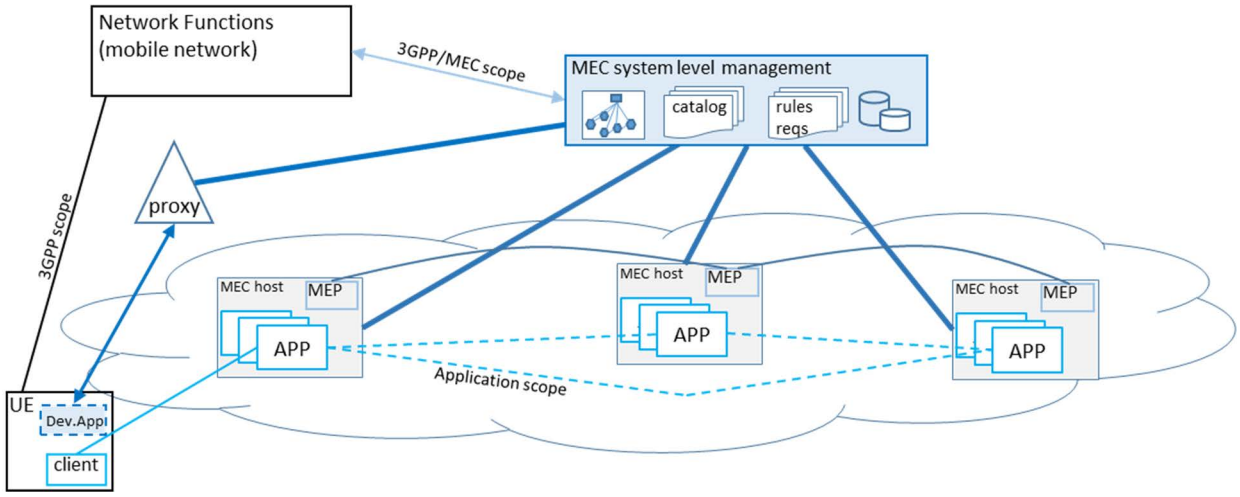
Annex A (informative): Key concepts

A.1 Application mobility

A.1.1 Introduction

MEC is a managed distributed cloud system consisting of system level, host level and network level functional entities. MEC orchestrator as the key system level management entity is in charge of maintaining an overall view of the MEC system, including the hosts, the physical resources, services and system topology. It is also in charge of selecting the MEC host for application instantiation (see ETSI GS MEC 003 [3]).

MEC application mobility includes the support for the transfer of the user context (i.e. user specific application state) from the source MEC host to the target MEC host, and the ability of the MEC system to instantiate the application in the target MEC host as a consequence of a UE mobility event. Figure A.1.1-1 illustrates the involved actors in MEC application mobility.



NOTE: For clarity, the MEC host level management has been omitted from figure A.1.1-1.

Figure A.1.1-1: Actors in MEC application mobility - Intra-MEC system scope

MEC system level management is connected to the mobile system e.g. to subscribe to the events of interest and to influence the routing in the data plane. MEC system level management, specifically the MEC orchestrator, is connected to all MEC hosts in the system (via host level management), and has the visibility and control over the hosts and can decide the instantiation and termination of the applications in those hosts.

NOTE 1: All interactions between the MEC system and the mobile system are currently not specified.

MEC Platforms (MEP) in different MEC hosts of the MEC system may be connected, provided that the MEPs support the platform-to-platform interface over Mp3 reference point. In the presence of Mp3 MEPs could exchange information via Mp3 to trigger the instantiation and the termination of the applications instances as part of application mobility support procedure.

NOTE 2: Platform-to-platform interface over Mp3 reference point is currently not specified.

A device application, if present in the UE, is connected to the user application lifecycle management proxy via Mx2 reference point, which in turn is connected to the MEC system level management for a specific MEC application's lifecycle management procedures. Via Mx2 to the device application may trigger the instantiation of the application instance as part of user application lifecycle management procedure.

The application client in the UE is connected to its server application instance in a MEC host. The application client in the UE is not expected to be edge aware, i.e. the client is agnostic to the deployment of the server application in the network edge.

Instances of the same MEC application in different MEC hosts may be connected to each other, provided that the application is designed for a distributed deployment and the MEC operator provides them with the connectivity.

A.1.2 Application availability in the target host

Application needs to be made available in the target MEC host in an event where the selected target does not have the desired application in place to serve the user(s) who is relocated to the said MEC host.

Once the target MEC host has been selected, the MEC orchestrator can determine if instantiation of the application is required or not. MEC orchestrator has the ability to bring the application image down to the selected MEC host and get it instantiated there by the Virtualization Infrastructure Manager (VIM).

Once an instance of the application is available on the selected target MEC host, the application mobility service sets up an interface of the communication link between the source application instance and the target application instance. The paired application instances then execute the user context transfer over that communication link.

A.1.3 User context transfer

For service continuity for a stateful application service, it is necessary to import the user context from the source application instance into the target application instance in the target MEC host.

As specified in clause 5.1, there are three high level implementation approaches for application state transfer:

1) device application assisted state transfer;

2) MEC assisted state transfer; and

3) application self-controlled state transfer.

The application state transfer is dependent on the capabilities of the application itself and of the underlying operating system of the MEC host. Both of these aspects are outside the scope of MEC specifications.

A.2 Mapping of permissions for RESTful API and resources

Table A.2-1 lists the permission categories for resources and topics in AMS API specification.

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

Permission identifier	Display name	Remarks
app_mobility_services	Application mobility service	Create, Query
deregister_task	Deregister application mobility service	Notify
adjacent_app_instances	Adjacent application instance information	Query
mobility_proc	Mobility procedure	Subscribe
adj_app_info	Adjacent application information	Subscribe

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

Table A.2-2: Permission identifiers mapping for resource

Permission identifier	Specification
app_mobility_services	Resource: …/amsi/v1/app_mobility_services
deregister_task	Resource: …/amsi/v1/app_mobility_services/{appMobilityServiceId}/deregister_task
adjacent_app_instances	Resource: …/amsi/v1/queries/adjacent_app_instances
mobility_proc	Resource: …/amsi/v1/subscriptions/mobility_proc
adj_app_info	Resource: …/amsi/v1/subscriptions/adj_app_info

History

Document history		
V2.1.1	January 2020	Publication