ETSI GS MEC 010-2 V2.1.1 (2019-11)

GROUP SPECIFICATION

Multi-access Edge Computing (MEC); MEC Management; Part 2: Application lifecycle, rules and requirements management

Disclaimer

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

Reference
RGS/MEC-0010-2v211LcRRM

Keywords
MEC

ETSI

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

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

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

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

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

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

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

Copyright Notification

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).

The present document is part 2 of a multi-part deliverable covering MEC Management, as identified below:

Part 1: "System, host and platform management";

Part 2: "Application lifecycle, rules and requirements management".

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 information flows for lifecycle management of applications running on a MEC host, and describes interfaces over the reference points to support application lifecycle management. It also describes application rules and requirements, application-related events, mobility handling and MEC service availability tracking. The present document specifies the necessary data model, data format and operation format when applicable.

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 NFV-IFA 011: "Network Functions Virtualisation (NFV); Management and Orchestration; VNF Descriptor and Packaging Specification".

[2] IETF RFC 4776: "Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Address Configuration Information".

[3] ISO 3166: "Codes for the representation of names of countries and their subdivisions".

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

[5] IETF RFC 8259: "The JavaScript Object Notation (JSON) Data Interchange Format".

NOTE: Available at https://tools.ietf.org/html/rfc8259.

[6] IETF RFC 7233: "Hypertext Transfer Protocol (HTTP/1.1): Range Requests".

[7] ETSI GS NFV-SOL 003: "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Or-Vnfm Reference Point".

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

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

[i.3] ETSI GS NFV-IFA 007: "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification".

[i.4] ETSI GS MEC 011: "Multi-access Edge Computing (MEC); Edge Platform Application Enablement".

[i.5] ETSI GS NFV 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

[i.6] ETSI GS NFV-SOL 004: "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; VNF Package specification".

[i.7] ETSI GS NFV-SOL 005: "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

application descriptor: descriptor provided by the application provider which describes the application rules and requirements of a MEC application

application package: bundle of files provided by application provider, on-boarded into MEC system and used by the MEC system for application instantiation, including an application descriptor, a VM image or a URI to a VM image, a manifest file, and other optional files

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GS MEC 001 [i.1], ETSI GS NFV 003 [i.5] and the following apply:

FQDN	Fully Qualified Domain Name
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
GTP	GPRS Tunnelling Protocol
GTP-U	GPRS Tunnelling Protocol for User plane
QCI	QoS Class Identifier
RNIS	Radio Network Information Service
URI	Uniform Resource Indicator

4 Specification level requirements

4.1 Requirements for reference point Mm1

4.1.1 General requirements

The Mm1 reference point between the MEC Orchestrator and the OSS is used for on-boarding application packages, triggering the instantiation and the termination of MEC applications in the MEC system. Table 4.1.1-1 specifies requirements related to application lifecycle management applicable to the Mm1 reference point. Those requirements are derived from table 5.2-1 of ETSI GS NFV-IFA 007 [i.3].

Table 4.1.1-1: Mm1 reference point requirements

Numbering	Functional requirement description
Mm1.001	The Mm1reference point shall support the application package management interface produced by the MEC Orchestrator.
Mm1.002	The Mm1reference point shall support the application lifecycle management interface produced by the MEC Orchestrator.

4.1.2 Interface requirements

4.1.2.1 Application package management interface requirements

Table 4.1.2.1-1 specifies requirements applicable to the application package management interface produced by the MEC Orchestrator on the Mm1 reference point. Those requirements are derived from clause 5.2 of ETSI GS MEC 002 [i.2].

Table 4.1.2.1-1: Application package management interface requirements

Numbering	Functional requirement description
Mm1.AppPkgm.001	The Application Package Management interface produced by the MEO on the Mm1 reference point shall support on-boarding an Application Package.
Mm1.AppPkgm.002	The Application Package Management interface produced by the MEO on the Mm1 reference point shall support querying Application Package information (see note).
Mm1.AppPkgm.003	The Application Package Management interface produced by the MEO on the Mm1 reference point shall support deleting an Application Package.
Mm1.AppPkgm.004	The Application Package Management interface produced by the MEO on the Mm1 reference point shall support enabling an application package.
Mm1.AppPkgm.005	The Application Package Management interface produced by the MEO on the Mm1 reference point shall support disabling an application package.
Mm1.AppPkgm.006	The Application Package Management interface produced by the MEO on the Mm1 reference point shall support providing notifications as a result of changes on application package states.
Mm1.AppPkgm.007	The Application Package Management interface produced by the MEO on the Mm1 reference point shall support providing notifications about the on-boarding of application packages.
Mm1.AppPkgm.008	The Application Package Management interface produced by the MEO on the Mm1 reference point shall support fetching an application package, or selected files contained in a package.
NOTE: Application package information may include information such as release date, vendor info, manifest, application descriptor, and other files contained in the application package, etc.	

4.1.2.2 Application lifecycle management interface requirements

Table 4.1.2.2-1 specifies requirements applicable to the application lifecycle management interface produced by the MEC Orchestrator on the Mm1 reference point.

Table 4.1.2.2-1: Application lifecycle management interface requirements

Numbering	Functional requirement description
Mm1.AppLcm.001	The Application Lifecycle Management interface produced by the MEO on the Mm1 reference point shall support instantiating an Application instance.
Mm1.AppLcm.002	The Application Lifecycle Management interface produced by the MEO on the Mm1 reference point shall support terminating an Application instance.
Mm1.AppLcm.003	The Application Lifecycle Management interface produced by the MEO on the Mm1 reference point shall support requesting to change the state of an application instance (see note).
Mm1.AppLcm.004	The Application Lifecycle Management interface produced by the MEO on the Mm1 reference point shall support providing notifications as a result of lcm operations.
NOTE: Changing the state of an application instance refers to starting or stopping an application instance. These operations are complementary to instantiating or terminating an application.	

4.2 Requirements for reference point Mm3

4.2.1 General requirements

The Mm3 reference point between the MEC Orchestrator and the MEC Platform Manager is used for the management of the application lifecycle, application rules and requirements and keeping track of available MEC services, etc. Table 4.2.1-1 specifies requirements related to application lifecycle management applicable to the Mm3 reference point.

Table 4.2.1-1: Mm3 reference point requirements

Numbering	Functional requirement description
Mm3.001	The Mm3 reference point shall support the application package management interface produced by the MEC Orchestrator.
Mm3.002	The Mm3 reference point shall support the application Lifecycle Management interface produced by the MEC Platform Manager.
Mm3.003	The Mm3 reference point shall support the application Lifecycle Change Notification interface produced by the MEC Platform Manager.

4.2.2 Interface requirements

4.2.2.1 Application package management interface requirements

Table 4.2.2.1-1 specifies requirements applicable to the Application Package Management interface produced by the MEO on the Mm3 reference point.

Table 4.2.2.1-1: Application Package Management interface requirements

Numbering	Functional requirement description
Mm3.AppPkgm.001	The Application Package Management interface produced by the MEO on the Mm3 reference point shall support querying application package information (see note).
Mm3.AppPkgm.002	The Application Package Management interface produced by the MEO on the Mm3 reference point shall support providing notifications as a result of changes on application package states.
Mm3.AppPkgm.003	The Application Package Management interface produced by the MEO on the Mm3 reference point shall support providing notifications about the on-boarding of application packages.
Mm3.AppPkgm.004	The Application Package Management interface produced by the MEO on the Mm3 reference point shall support fetching an application package, or selected files contained in a package.
NOTE: Application package information may include information such as release date, vendor info, manifest, application descriptor, and other files contained in the application package, etc.	

4.2.2.2 Application lifecycle management interface requirements

Table 4.2.2.2-1 specifies requirements applicable to the Application Lifecycle Management interface produced by the MEC Platform Manager on the Mm3 reference point.

Table 4.2.2.2-1: Application Lifecycle Management interface requirements

Numbering	Functional requirement description
Mm3.AppLcm.001	The Application Lifecycle Management interface produced by the MEC Platform Manager on the Mm3 reference point shall support instantiating an Application.
Mm3.AppLcm.002	The Application Lifecycle Management interface produced by the MEC Platform Manager on the Mm3 reference point shall support terminating an application instance.
Mm3.AppLcm.003	The Application Lifecycle Management interface produced by the MEC Platform Manager on the Mm3 reference point shall support querying information about an application instance.
Mm3.AppLcm.004	The Application Lifecycle Management interface produced by the MEC Platform Manager on the Mm3 reference point shall support requesting to change the state of an application instance (see note).
Mm3.AppLcm.005	The Application Lifecycle Management interface produced by the MEC Platform Manager on the Mm3 reference point shall support querying the status of an ongoing application lifecycle management operation.
NOTE: Changing the state of an application instance refers to starting or stopping an application instance. These operations are complementary to instantiating or terminating an application.	

4.2.2.3 Application lifecycle change notification interface requirements

Table 4.2.2.3-1 specifies requirements applicable to the Application Lifecycle Change Notification interface produced by the MEC Platform manager on the Mm3 reference point.

Table 4.2.2.3-1: Application Lifecycle Change Notification interface requirements

Numbering	Functional requirement description
Mm3.AppLccn.001	The Application Lifecycle Change Notification interface produced by the MEC Platform Manager on the Mm3 reference point shall support providing to the MEO notifications about changes of an application instance that are related to application lifecycle management operations.
Mm3.AppLccn.002	Notifications provided on the Application Lifecycle Change Notification interface produced by the MEC Platform Manager on the Mm3 reference point shall contain information about the type of application lifecycle operation, the identification of the application instance.
Mm3.AppLccn.003	Notifications provided on the Application Lifecycle Change Notification interface produced by the MEC Platform Manager on the Mm3 reference point shall support indicating the start of the lifecycle operation, the end and the results of the lifecycle operation including any error produced from the lifecycle operation.
Mm3.AppLccn.004	The Application Lifecycle Change Notification interface produced by the MEC Platform Manager on the Mm3 reference point shall support notifying the result (successful or failed) of application instantiation with indicating the application instance identifier, and the consumed, modified or released resources.

4.3 Requirements for application package

4.3.1 General requirements

Table 4.3.1-1 specifies requirements related to application lifecycle management applicable to the application package.

Table 4.3.1-1: Application package requirements

Numbering	Functional requirement description
AppPkt.001	The application package shall contain software image(s) or link(s) to software image(s). See note.
AppPkt.002	The application package shall contain an application descriptor that describes the application requirements and rules which are required or preferred by the MEC application. See note.
AppPkt.003	The application package shall be signed by the application provider. The digest and the public key of the entity signing shall be included in the package along with the corresponding certificate.

Numbering	Functional requirement description
AppPkt.004	Files in the package may be individually signed. For each signed file, the corresponding public key, algorithm and certificate used shall be stored in a well-known location within the application package.
AppPkt.005	The application package shall contain a manifest file which lists files that the package contains and a hash of their content. See note.
NOTE: The application package, the software image(s), the manifest file and the application descriptor are provided by the application provider.	

4.3.2 Application descriptor requirements

Table 4.3.2-1 specifies requirements related to application lifecycle management applicable to the application descriptor.

Table 4.3.2-1: Application descriptor requirements

Numbering	Functional requirement description
AppDesc.001	The application descriptor shall contain a description of minimum computation resources required by the application, e.g. amount, characteristics and capabilities for virtual compute.
AppDesc.002	The application descriptor shall contain a description of minimum virtual storage resources the required by application.
AppDesc.003	The application descriptor shall contain a description of minimum virtual network resources required by the application.
AppDesc.004	The application descriptor shall support describing a list of services a MEC application requires to run.
AppDesc.005	The application descriptor shall support describing a list of additional services that a MEC application may use if available.
AppDesc.006	The application descriptor shall support describing a list of features a MEC application requires to run.
AppDesc.007	The application descriptor shall support describing a list of additional features a MEC application may use if available.
AppDesc.008	The application descriptor shall support a description of Traffic Rules.
AppDesc.009	The application descriptor shall support a description of DNS Rules which provide specific FQDNs to be registered into the MEC system (e.g. for redirection of traffic to local host).
AppDesc.010	The application descriptor shall support a description of latency required by the MEC application.

5 Message flows to support application package and lifecycle management

5.1 General

Message flows in this clause are informative.

5.2 Application package management

5.2.1 General

Message flows for application package management are used to make application package available to the MEC system, delete the application package from the MEC system, or query information of one or more application packages. The series of message flows include:

- On-board application package.
- Query application package information.
- Disable application package.
- Enable application package.

- Delete application package.
- Fetch application package.

This interface is available on the reference points Mm1 and Mm3. On Mm1, the full functionality of this interface is supported. On Mm3, only read access to the packages and their content is supported.

5.2.2 On-board application package

The message flow of on-board application package is used to make application package available to the MEC system. The on-board application package message flow is executed before an application is instantiated; the actual time to execute this message flow is dependent on implementation. The detailed description of the flow is in figure 5.2.2-1.

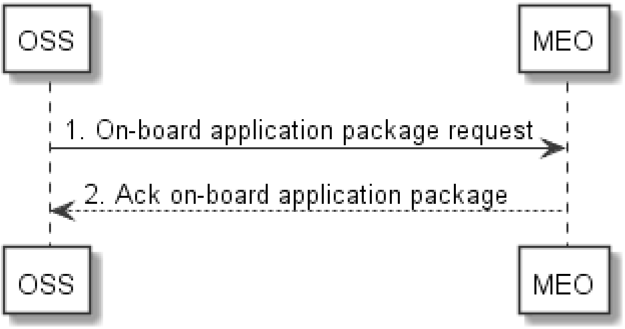
OSS
MEO
1. On-board application package request
2. Ack on-board application package
OSS
MEO

Figure 5.2.2-1: On-board application package

1) The OSS sends an on-board application package request to the MEC Orchestrator, in which, the MEC application package is included.

   The MEC Orchestrator checks the application package, for example: the MEC Orchestrator checks for the existence of mandatory elements within the application package, validates the authenticity and integrity of the application package; and the MEC Orchestrator checks the format of application image and format of the application rules and requirements.

   The MEC Orchestrator allocates a unique application package ID for the on-boarded MEC application package and related status information, and keeps a record of on-boarded application packages. Optionally the MEC Orchestrator prepares the virtualisation infrastructure manager(s) with the application image (e.g. by sending the application image to appropriate virtualisation infrastructure managers), alternatively, such preparation may be done later, but needs to be finished before the application is instantiated. The MEC Orchestrator notifies the subscribers to AppPackageOnBoardingNotification of the on-boarding of the MEC application package.

2) The MEC Orchestrator acknowledges the application package on-boarding to the OSS. The application package is now available in the MEC system and enabled.

This message flow is only supported on Mm1.

5.2.3 Query application package information

Query application package allows returning the information contained in an application package (such as the location of the application descriptor) to the OSS and the MEPM. The detailed description of the flow is in figure 5.2.3-1.

This flow can be used by the MEPM to obtain the application requirements, traffic redirection rules and DNS rules that it needs to configure the MEC platform to run the application. This information is provided as defined by the application provider in the application package. Modification of these rules by the MEO prior to sending it to the MEPM is not supported in the present version of the present document.

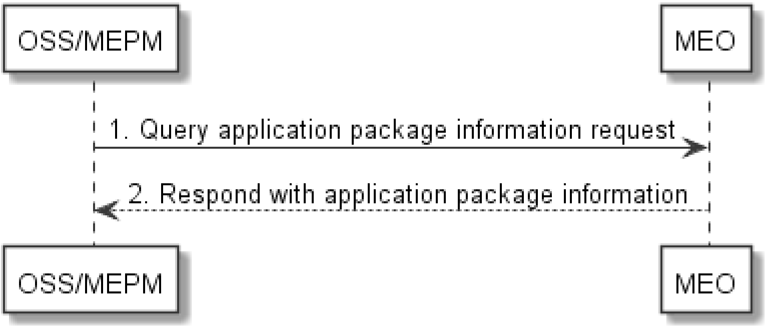
OSS/MEPM
MEO
1. Query application package information request
2. Respond with application package information
OSS/MEPM
MEO

Figure 5.2.3-1: Query application package information

1) The OSS or MEPM sends a query application package information request (including a filter) to the MEC Orchestrator to query information of application package(s).

   The MEC Orchestrator authorizes the request.

2) The MEC Orchestrator returns the information related to the application package(s) that matches the filter in the query request.

This message flow is supported on Mm1 and Mm3.

5.2.4 Disable application package

Disabling application package refers to the process of marking an application package as disabled in the MEC system, so that it is not possible to be used for application instantiation. The detailed description of the flow is in figure 5.2.4-1.

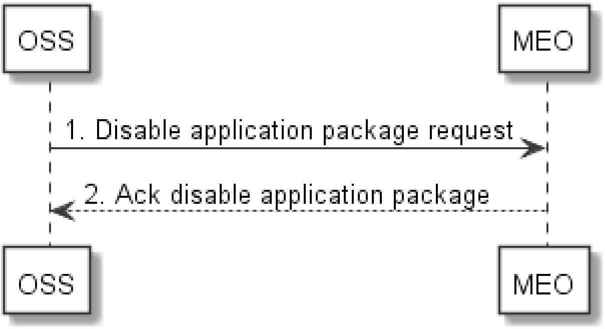
OSS
MEO
1. Disable application package request
2. Ack disable application package
OSS
MEO

Figure 5.2.4-1: Disable application package

1) The OSS sends a request to disable application package to the MEC Orchestrator.

   The MEC Orchestrator processes the request and checks if the application package exists, and is enabled. If the application package is enabled, the MEC Orchestrator marks the application package as disabled in the MEC system.

   The MEC Orchestrator notifies the subscribers to AppPackageStateChangeNotification of the state change of the MEC application package.

2) The MEC Orchestrator acknowledges the disable application package request.

This message flow is only supported on Mm1.

5.2.5 Enable application package

Enabling an application package refers to the process of marking an application package as enabled in the MEC system, so that it may be used for application instantiation again. The detailed description of the flow is in figure 5.2.5-1.

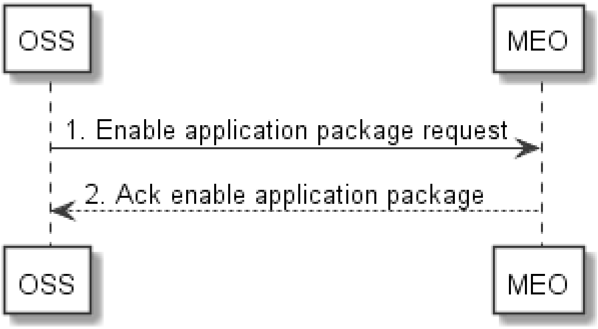
OSS
MEO
1. Enable application package request
2. Ack enable application package
OSS
MEO

Figure 5.2.5-1: Enable application package

1) The OSS sends an enable application package request to the MEC Orchestrator.

   The MEC Orchestrator processes the request and checks if the application package exists, and is disabled. If the application package is disabled, the MEC Orchestrator marks the application package as enabled in the MEC system.

   The MEC Orchestrator notifies the subscribers to AppPackageStateChangeNotification of the state change of the MEC application package.

2) The MEC Orchestrator acknowledges the application package enable request.

This message flow is only supported on Mm1.

5.2.6 Delete application package

Delete application package refers to the process of removing an application package from the MEC system. The OSS initiates the delete application package message flow. The detailed description of the flow is in figure 5.2.6-1.

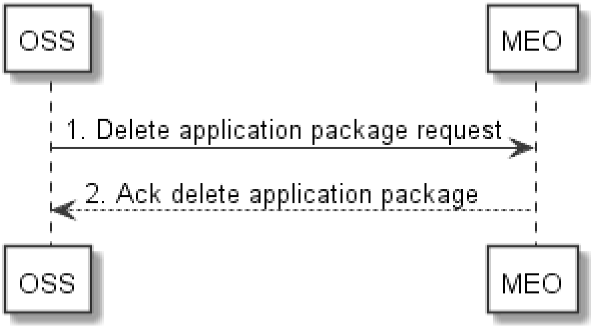
OSS
MEO
1. Delete application package request
2. Ack delete application package
OSS
MEO

Figure 5.2.6-1: Delete application package

1) The OSS sends a delete application package request to the MEC Orchestrator.

   The MEC Orchestrator checks whether the application package is onboarded, disabled and in use. If the application package is not onboarded, not disabled or is in use, this operation will return an error. If the application package is disabled and is not in use, the MEC Orchestrator removes the application package from the MEC system.

   The MEC Orchestrator notifies the subscribers to AppPackageStateChangeNotification of the state change of the MEC application package.

2) The MEC Orchestrator acknowledges application package deletion request.

This message flow is only supported on Mm1.

5.2.7 Fetch onboarded application package

Fetch onboarded application package allows retrieving an application package, or selected files contained in a package to the OSS and the MEPM. The detailed description of the flow is in figure 5.2.7-1.

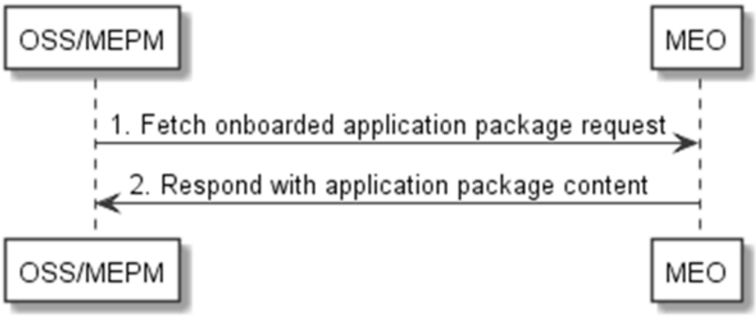
OSS/MEPM
MEO
1. Fetch onboarded application package request
2. Respond with application package content
OSS/MEPM
MEO

Figure 5.2.7-1: Fetch onboarded application package

1) The OSS or MEPM sends a fetch onboarded application package request to the MEC Orchestrator to fetch the content of an application package.

   The MEC Orchestrator authorizes the request.

2) The MEC Orchestrator returns the application package.

5.3 Application instance lifecycle management

5.3.1 Application instantiation

The message flow of application instantiation is used to instantiate an application instance in the MEC system. The detailed description of the flow is in figure 5.3.1-1.

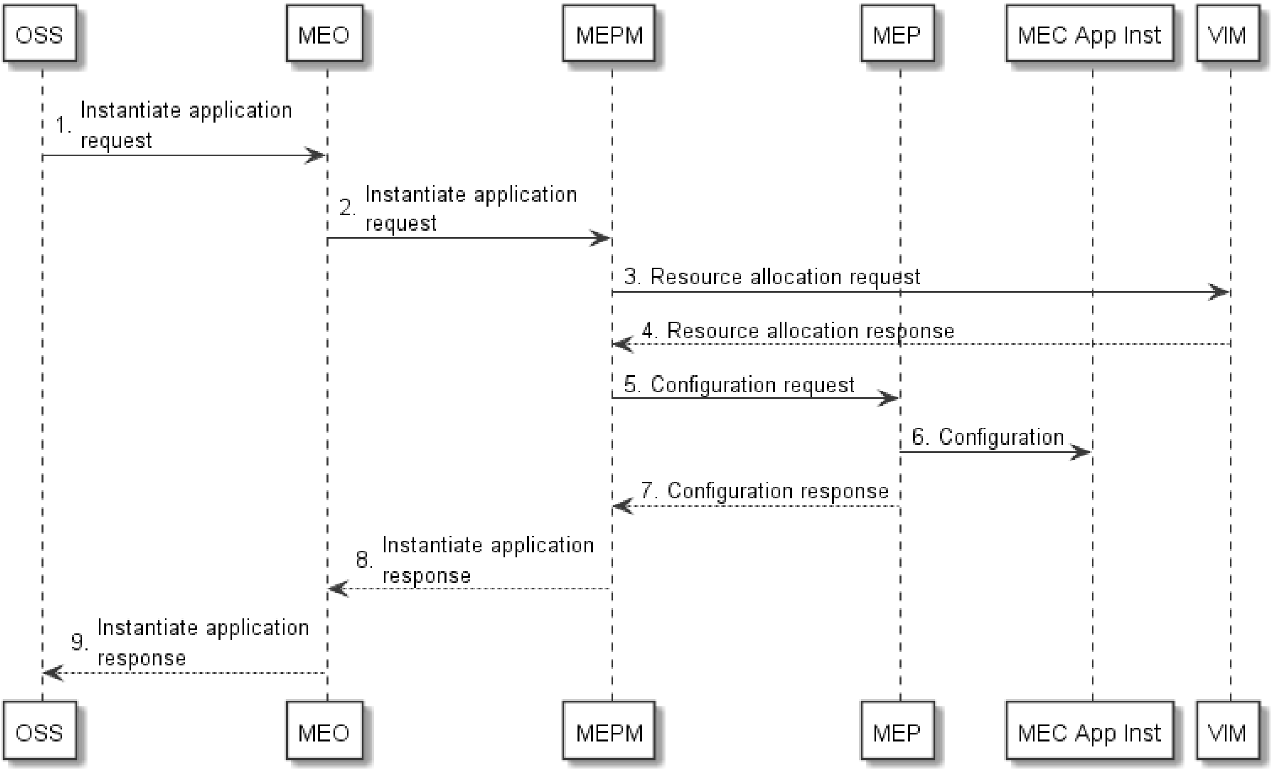
OSS
MEO
MEPM
MEP
MEC App Inst
VIM
1. Instantiate application request
2. Instantiate application request
3. Resource allocation request
4. Resource allocation response
5. Configuration request
6. Configuration
7. Configuration response
8. Instantiate application response
9. Instantiate application response
OSS
MEO
MEPM
MEP
MEC App Inst
VIM

Figure 5.3.1-1: Application instantiation flow

1) The OSS sends an instantiate application request to the MEC Orchestrator.

2) MEC Orchestrator checks the application instance configuration data, and authorizes the request. The MEC Orchestrator selects the MEC host (and corresponding MEC Platform Manager), and sends an instantiate application request to the MEC Platform Manager.

3) The MEC Platform Manager sends a resource allocation request to the Virtualisation Infrastructure Manager (VIM), with the requested resource including compute, storage, and network resources. The MEC Platform Manager will include application image information (e.g. a link to the image or an ID of the application image) in the request.

4) The VIM allocates the resources according to the request of the MEC Platform Manager. And if the application image is available, the VIM loads the virtual machine with the application image, and runs the VM and the application instance. The VIM sends resource allocation response to the MEC Platform Manager.

NOTE 1: Optionally, if traffic redirection is realized based on NFVI mechanisms controlled by the VIM, the MEC Platform Manager further determines the need to create or update a forwarding path based on the traffic rule(s) (associated with the instantiated application), and sends a forwarding path creation or update request to the VIM. The VIM creates or updates the forwarding path according to the indication of the request and sends a forwarding path creation or update response to the MEC Platform Manager. The MEC Platform Manager associates the forwarding path with the traffic rule(s) of the instantiated application.

NOTE 2: If traffic redirection is realized based on mechanisms internal to the data plane and controlled via Mp2, it is out of scope of the present document how traffic redirection via Mp2 during app creation, and during enabling, disabling or modifying a traffic rule are controlled.

5) The MEC Platform Manager sends configuration request to the MEC platform. In this message, the MEC Platform Manager includes the traffic rules to be configured, DNS rules to be configured, the required and optional services, and services produced by the application instance, etc.

NOTE 3: This step is only for the completeness of the flow, and will not be specified in the present document.

6) The MEC platform configures the Traffic rules and DNS rules for the application instance. The MEC platform needs to wait until the application instance runs normally (e.g. the application instance state turns into the running state) to activate the traffic and DNS rules. After the application instance runs normally, the MEC platform provides the available service information to the application. The details of how the MEC application instance interacts with MEC platform and how the MEC platform handles services can be found in ETSI GS MEC 011 [i.4].

7) The MEC platform sends a configuration response to the MEC Platform Manager.

NOTE 4: This step is only for the completeness of the flow, and will not be specified in the present document.

8) The MEC Platform Manager sends an instantiate application response to the MEC Orchestrator. The MEC Platform Manager includes the information of the resources allocated to the application instance to the MEC Orchestrator.

9) The MEC Orchestrator sends an instantiate application response to the OSS, returning the results of the instantiation operation. The MEC Orchestrator also returns the application instance ID to the OSS if the flow is successful. Meanwhile a notification is sent to the subscribers to the application instance operational state change notification and this application package is marked in use.

5.3.2 Application termination

The message flow of application instance termination is used to terminate an application instance in the MEC system. The detailed description of the flow is in figure 5.3.2-1.

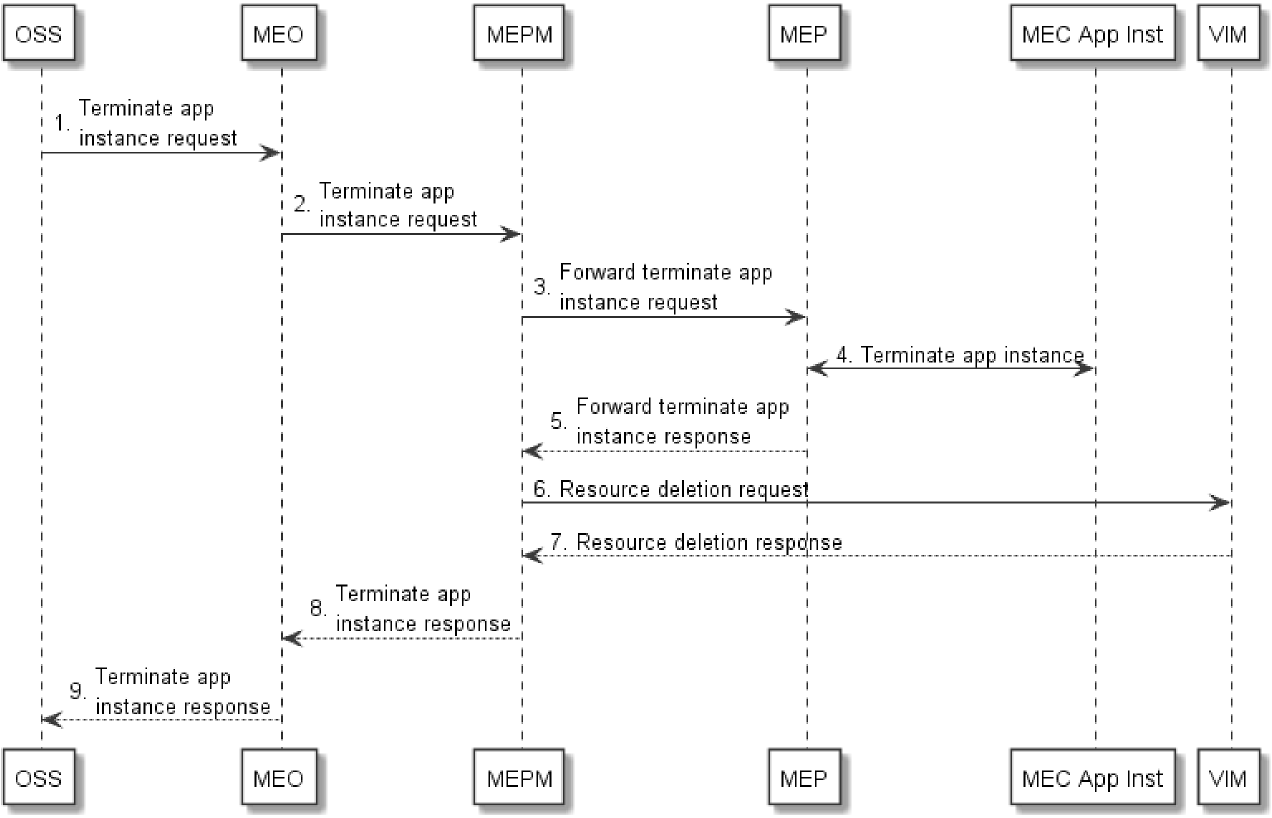
OSS
MEO
MEPM
MEP
MEC App Inst
VIM
1. Terminate app instance request
2. Terminate app instance request
3. Forward terminate app instance request
4. Terminate app instance
5. Forward terminate app instance response
6. Resource deletion request
7. Resource deletion response
8. Terminate app instance response
9. Terminate app instance response

Figure 5.3.2-1: Instance Termination information flow

1) The OSS sends a terminate application instance request to the MEC Orchestrator. The message includes the MEC application instance ID to be terminated.

2) The MEC Orchestrator authorizes the request, and verifies the existence of requested instance(s). The MEC Orchestrator sends a terminate application instance request to the MEC Platform Manager that manages the MEC application instance(s) to be terminated.

3) The MEC Platform Manager sends a forward terminate application instance request to the MEC platform.

NOTE 1: This step is only for the completeness of the flow, and will not be specified in the present document.

4) If supported by the MEC application, and graceful termination is requested, the MEC platform notifies the MEC application instance of the termination event. The MEC application instance executes the actions needed before it has been terminated by the MEC platform, the actual action(s) the MEC application instance will perform for the application level termination is up to MEC application, and is out of scope of the present document. After the MEC application instance finishes application level termination, it may inform the MEC platform that it is ready to be terminated. The MEC platform may set a timer for the application level termination. After the timer expires, the MEC platform will shut down the application regardless of the progress of application level termination.

5) The MEC platform sends a forward terminate application instance response to the MEC Platform Manager.

NOTE 2: This step is only for the completeness of the flow, and will not be specified in the present document.

6) The MEC Platform Manager sends a resource deletion request to the corresponding virtualisation infrastructure manager, to terminate the virtual machine and release the resources.

7) The virtualisation infrastructure manager releases the resources allocated for the application instance to be terminated. And the virtualisation infrastructure manager sends resource a deletion response to the MEC Platform Manager.

8) The MEC Platform Manager sends a terminate application instance response to the MEC Orchestrator.

9) The MEC Orchestrator sends a terminate application instance response to the OSS. Meanwhile a notification is sent to the subscribers to the application instance operational state change notification.

5.3.3 Application operation

This message flow is to operate (i.e. start or stop) an application instance in the MEC system. The detailed description of the flow is in figure 5.3.3-1.

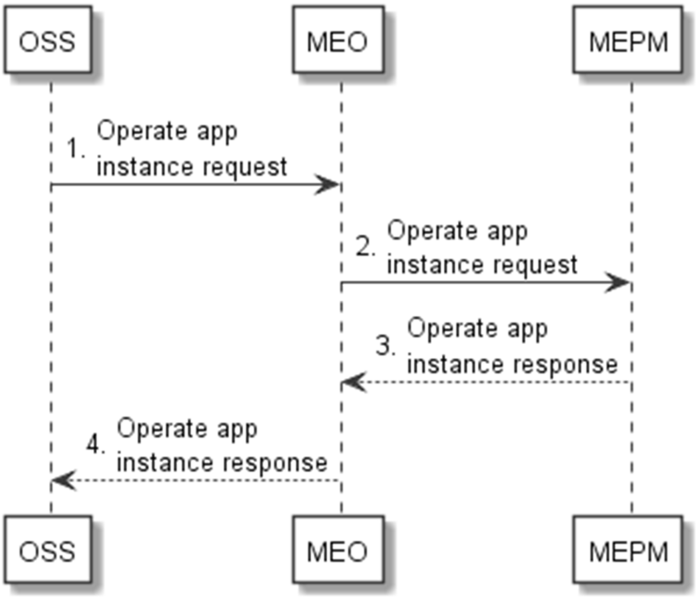
OSS
MEO
MEPM
1. Operate app instance request
2. Operate app instance request
3. Operate app instance response
4. Operate app instance response
OSS
MEO
MEPM

Figure 5.3.3-1: Application operation flow

1) The OSS sends an operate (i.e. start or stop) application request to the MEC Orchestrator.

2) The MEC Orchestrator forwards the operate application request to the MEC Platform Manager.

3) The MEC Platform Manager processes this request, and sends the result of operation on the application instance once the operation completes.

4) The MEC Orchestrator sends the result of application operation to the OSS.

5.4 Invoking application LCM operations

This clause describes the general sequence for application instance lifecycle management operations that operate on application instance resource and are triggered by task resources. The operations of application instance lifecycle management are:

- Instantiate.
- Operate.
- Terminate.

The MEO can use a resource that represents an application instance LCM operation occurrence to monitor the progress of the operation. The MEO can also subscribe to notifications sent by the MEPM when an application instance LCM operation occurrence changes its state. Further, each application LCM operation invocation triggers a granting procedure between MEPM and MEO.

Table 5.4-1 shows parameterization associated to above application instance lifecycle management operations.

Table 5.4-1: Parameterization for application instance lifecycle management operations

Operation	Precondition	Task	RequestStructure	Postcondition
Instantiate	Application instance created and in NOT_INSTANTIATED state	instantiate	InstantiateAppRequest	Application instance in INSTANTIATED state
Operate	Application instance in INSTANTIATED state	operate	OperateAppRequest	Application instance still in INSTANTIATED state and application instance operational state changed
Terminate	Application instance in INSTANTIATED state	terminate	TerminateAppRequest	Application instance in NOT_INSTANTIATED state

Figure 5.4-1 illustrates the general lifecycle management flow with granting procedure for the application lifecycle management. Placeholders in this flow allow for differentiating among the operations and are marked with double angular brackets "<<…>>".

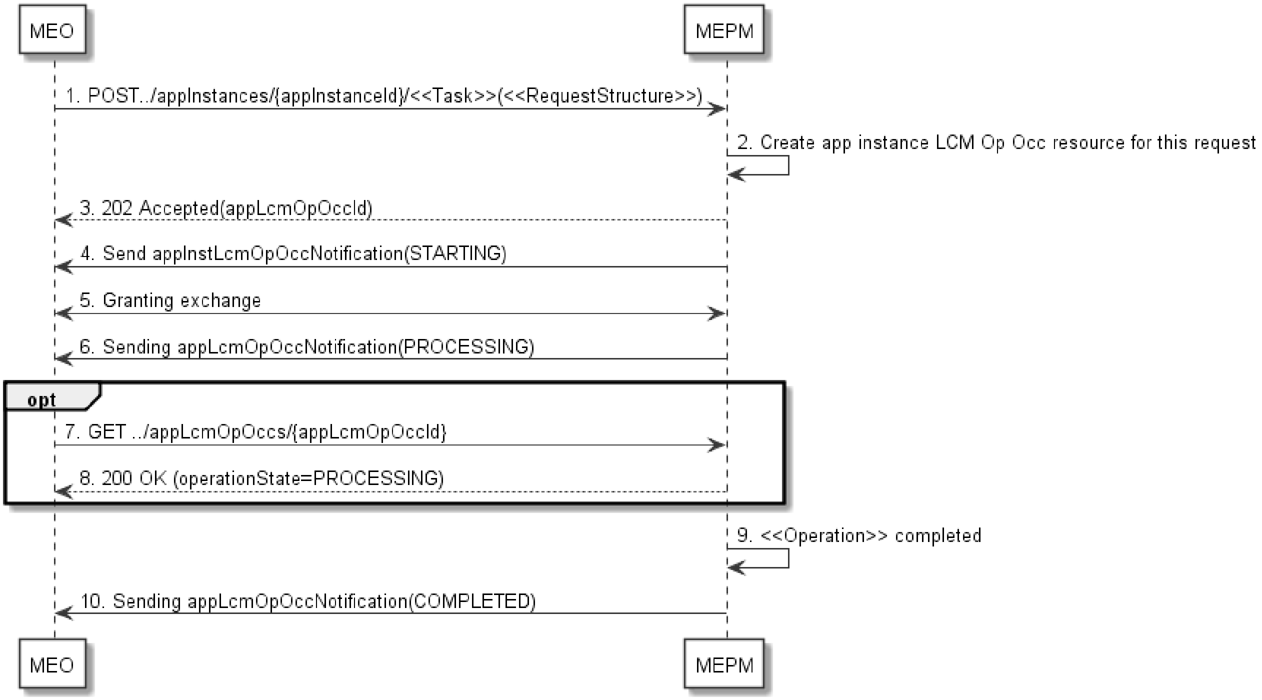
MEO
MEPM
1. POST../appInstances/{appInstanceId}/<<Task>>(<<RequestStructure>>)
2. Create app instance LCM Op Occ resource for this request
3. 202 Accepted(appLcmOpOccId)
4. Send appInstLcmOpOccNotification(STARTING)
5. Granting exchange
6. Sending appLcmOpOccNotification(PROCESSING)
opt
7. GET ../appLcmOpOccs/{appLcmOpOccId}
8. 200 OK (operationState=PROCESSING)
9. <<Operation>> completed
10. Sending appLcmOpOccNotification(COMPLETED)
MEO
MEPM

Figure 5.4-1: Application instance LCM operation with granting procedure

An application instance lifecycle operation, as illustrated in figure 5.4-1, consists of the following steps:

1) The MEO sends a POST request to the <<Task>> resource that represents the lifecycle operation to be executed on the application instance, and includes in the payload body a data structure of type <<RequestStructure>>. The <<Task>> of the task resource and the <<RequestStructure>> depend on the operation and are described in table 5.4-1.

2) The MEPM creates an "application instance LCM operation occurrence" resource for the request.

3) The MEPM returns a "202 Accepted" response with an empty payload body and a "Location" HTTP header that points to the new "application instance LCM operation occurrence" resource, i.e. it includes the URI of that resource which is "…/app_lcm_op_occs/{appLcmOpOccId}".

4) The MEPM sends to the MEO an application instance lifecycle management operation occurrence notification to indicate the start of the lifecycle management operation occurrence with the "STARTING" state.

5) The MEPM and MEO exchange granting information.

6) The MEPM sends to the MEO an application instance lifecycle management operation occurrence notification to indicate that the application instance LCM operation occurrence enters the "PROCESSING" state.

7) Optionally, the MEO may query the "application instance LCM operation occurrence" resource to obtain information about the ongoing operation by sending a GET request to the resource that represents the application instance LCM operation occurrence.

8) In the response to that request, the MEPM returns to the MEO information of the operation, such as the operation status in "AppLcmOpOcc".

9) The MEPM finishes the <<Operation>>.

10) The MEPM sends an application instance lifecycle management operation occurrence notification to indicate the completion of the lifecycle management operation occurrence with the success state "COMPLETED".

Figure 5.4-2 illustrates the synchronous granting procedure for an application lifecycle operation.

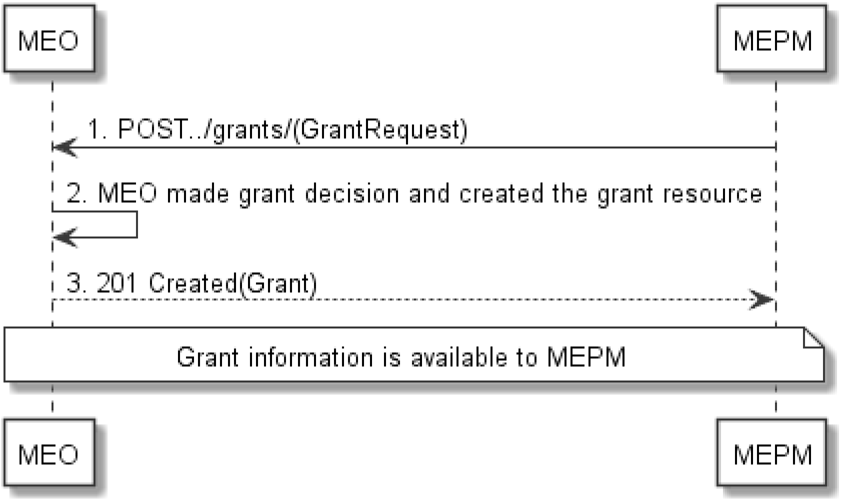
MEO
MEPM
1. POST../grants/(GrantRequest)
2. MEO made grant decision and created the grant resource
3. 201 Created(Grant)
Grant information is available to MEPM
MEO
MEPM

Figure 5.4-2: Synchronized granting procedure for application instance LCM operation

Granting with synchronous response, as illustrated in the figure 5.4-2, consists of the following steps:

1) The MEPM sends a POST request to the "Grants" resource with a "GrantRequest" data structure in the body.

2) The MEO makes the granting decision, and creates a new "Individual grant" resource.

3) The MEO returns to the MEPM a "201 Created" response with a "Grant" data structure in the body and a "Location" HTTP header that points to the new "Individual grant" resource.

Figure 5.4-3 illustrates the asynchronous granting procedure for an application lifecycle operation.

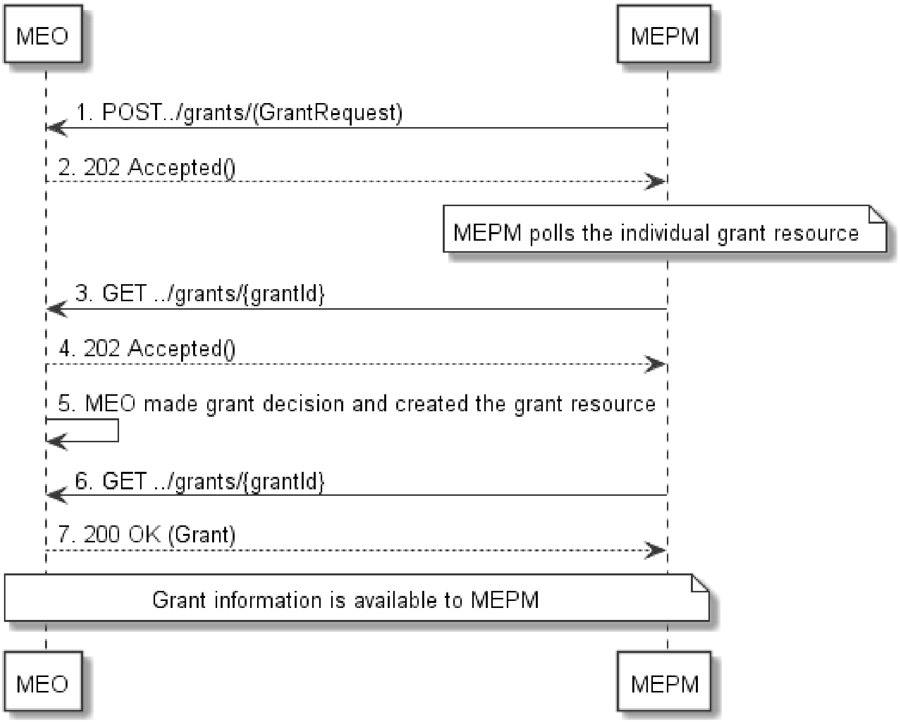
MEO
MEPM
1. POST../grants/(GrantRequest)
2. 202 Accepted()
MEPM polls the individual grant resource
3. GET ../grants/{grantId}
4. 202 Accepted()
5. MEO made grant decision and created the grant resource
6. GET ../grants/{grantId}
7. 200 OK (Grant)
Grant information is available to MEPM
MEO
MEPM

Figure 5.4-3: Asynchronous granting procedure for application instance LCM operation

Granting with asynchronous response, as illustrated in figure 5.4-3, consists of the following steps:

1) The MEPM sends a POST request to the "Grants" resource with a "GrantRequest" data structure in the body.

2) The MEO sends an Accepted response with return code 202 that contains the resource URI of the to-be-created "Individual grant" resource in the "Location" header.

3) MEPM may query the status of granting process via the GET method.

4) MEO sends an accepted response with return code 202 to indicate that the granting decision has not been made.

5) The MEO makes the granting decision, and creates a new "Individual grant" resource.

6) The MEPM may query the status of granting process via GET method.

7) This time the MEO responds with the return code 200 to confirm the grant.

6 Information models and interfaces

6.1 Applicable reference points

The following clauses apply to the Mm reference points, for which the relevant sequence diagrams are described in clause 5.

6.2 Information models

6.2.1 Application descriptor information model

6.2.1.1 Introduction

This clause defines data structures to be used by application descriptor information model.

6.2.1.2 Type: AppD

6.2.1.2.1 Description

An application Descriptor (AppD) is a part of application package, and describes application requirements and rules required by application provider.

6.2.1.2.2 Attributes

The attributes of the AppD data type shall follow the indications provided in table 6.2.1.2.2-1.

Table 6.2.1.2.2-1: Attributes of the AppD

Attribute name	Cardinality	Data type	Description
appDId	1	String	Identifier of this MEC application descriptor. This attribute shall be globally unique. See note 1.
appName	1	String	Name to identify the MEC application.
appProvider	1	String	Provider of the application and of the AppD.
appSoftVersion	1	String	Identifies the version of software of the MEC application.
appDVersion	1	String	Identifies the version of the application descriptor.
mecVersion	1..N	String	Identifies version(s) of MEC system compatible with the MEC application described in this version of the AppD.
appInfoName	0..1	String	Human readable name for the MEC application.
appDescription	1	String	Human readable description of the MEC application.
virtualComputeDescriptor	1	VirtualComputeDescription	Describes CPU, Memory and acceleration requirements of the virtual machine.
swImageDescriptor	1	SwImageDescriptor	Describes the software image which is directly loaded on the virtualisation machine instantiating this Application.
virtualStorageDescriptor	0..N	VirtualStorageDescriptor	Defines descriptors of virtual storage resources to be used by the MEC application.
appExtCpd	0..N	AppExternalCpd	Describes external interface(s) exposed by this MEC application.
appServiceRequired	0..N	ServiceDependency	Describes services a MEC application requires to run.
appServiceOptional	0..N	ServiceDependency	Describes services a MEC application may use if available.
appServiceProduced	0..N	ServiceDescriptor	Describes services a MEC application is able to produce to the platform or other MEC applications. Only relevant for service-producing apps.
appFeatureRequired	0..N	FeatureDependency	Describes features a MEC application requires to run.
appFeatureOptional	0..N	FeatureDependency	Describes features a MEC application may use if available.

Attribute name	Cardinality	Data type	Description
transportDependencies	0..N	TransportDependency	Transports, if any, that this application requires to be provided by the platform. These transports will be used by the application to deliver services provided by this application. Only relevant for service-producing apps. See note 2.
appTrafficRule	0..N	TrafficRuleDescriptor	Describes traffic rules the MEC application requires.
appDNSRule	0..N	DNSRuleDescriptor	Describes DNS rules the MEC application requires.
appLatency	0..1	LatencyDescriptor	Describes the maximum latency tolerated by the MEC application.
terminateAppInstanceOpConfig	0..1	TerminateAppInstanceOpConfig	Configuration parameters for the Terminate application instance operation.
changeAppInstanceStateOpConfig	0..1	ChangeAppInstanceStateOpConfig	Configuration parameters for the change application instance state operation.
NOTE 1: The appDId shall be used as the unique identifier of the application package that contains this AppD. NOTE 2: This attribute indicates groups of transport bindings which a service-producing MEC application requires to be supported by the platform in order to be able to produce its services. At least one of the indicated groups needs to be supported to fulfil the requirements.			

6.2.1.3 Type: VirtualComputeDescriptor

6.2.1.3.1 Description

The VirtualComputeDescriptor data type supports the specification of requirements related to virtual compute resources.

6.2.1.3.2 Attributes

The attributes of VirtualComputeDescription shall follow the definition in clause 7.1.9.2.2 of ETSI GS NFV-IFA 011 [1], with the following consideration:

- The VNF refers to MEC Application, the VNFD refers to AppD in MEC, and the VDU in table 7.1.9.5.2-1 of ETSI GS NFV-IFA 011 [1] refers to MEC application.

6.2.1.4 Type: SwImageDescriptor

6.2.1.4.1 Description

The SwImageDescriptor data type describes a software image of a MEC application.

6.2.1.4.2 Attributes

The attributes of SwImageDescriptor shall follow the definition in clause 7.1.6.5 of ETSI GS NFV-IFA 011 [1], with the following consideration:

- The VNF refers to MEC Application, the VNFD refers to AppD in MEC, and the VDU refers to MEC application.

6.2.1.5 Type: VirtualStorageDescriptor

6.2.1.5.1 Description

The VirtualStorageDescriptor data type describes a virtual storage required by a MEC application.

6.2.1.5.2 Attributes

The attributes of VirtualStorageDescriptor shall follow the definition in clause 7.1.9.4 of ETSI GS NFV-IFA 011 [1], with the following consideration:

- The VNF refers to MEC application, the VNFD refers to AppD in MEC, and the VDU refers to MEC application if they appear in ETSI GS NFV-IFA 011 [1].

6.2.1.6 Type: AppExternalCpd

6.2.1.6.1 Description

The AppExternalCpd data type supports the specification of MEC application requirements related to external connection point.

6.2.1.6.2 Attributes

The attributes of AppExternalCpd is shown in table 6.2.1.6.2-1.

Table 6.2.1.6.2-1: Attributes of the AppExternalCpd

Attribute name	Cardinality	Data type	Description
virtualNetworkInterfaceRequirements	0..N	VirtualNetworkInterfaceRequirements	Specifies requirements on a virtual network interface realizing the CPs instantiated from this CPD.
(inherited attributes)			All attributes inherited from Cpd.

For the definition of the VirtualNetworkInterfaceRequirements, please refer to clause 7.1.6.6 of ETSI GS NFV-IFA 011 [1]. For the definition of Cpd, please refer to clause 7.1.6.3 of ETSI GS NFV-IFA 011 [1].

6.2.1.7 Type: ServiceDescriptor

6.2.1.7.1 Description

The ServiceDescriptor data type describes a MEC service produced by a service-providing MEC application.

6.2.1.7.2 Attributes

The attributes of a ServiceDescriptor are depicted in table 6.2.1.7.2-1.

Table 6.2.1.7.2-1: Attributes of ServiceDescriptor

Attribute name	Cardinality	Data type	Description
serName	1	String	The name of the service, for example, RNIS, LocationService, etc.
serCategory	0..1	CategoryRef	A Category reference of the service, defined in ETSI GS MEC 011 [i.4].
version	1	String	The version of the service.
transportsSupported	0..N	Structure (inlined)	Indicates transports and serialization formats supported made available to the service-consuming application. Defaults to REST + JSON if absent.
>transport	1	TransportDescriptor	Information about the transport in this binding.
>serializers	1..N	SerializerTypes	Information about the serializers in this binding, as defined in the SerializerTypes type in ETSI GS MEC 011 [i.4].

6.2.1.8 Type: FeatureDependency

6.2.1.8.1 Description

The FeatureDependency data type supports the specification of requirements of a MEC application related to a feature of MEC platform.

6.2.1.8.2 Attributes

The attributes of a FeatureDependency are depicted in table 6.2.1.8.2-1.

Table 6.2.1.8.2-1: Attributes of FeatureDependency

Attribute name	Cardinality	Data type	Description
featureName	1	String	The name of the feature, for example, UserApps, UEIdentity, etc.
version	1	String	The version of the feature.

6.2.1.9 Type: TrafficRuleDescriptor

6.2.1.9.1 Description

The TrafficRuleDescriptor data type describes traffic rules related to a MEC application.

6.2.1.9.2 Attributes

The attributes of TrafficRuleDescriptor is shown in table 6.2.1.9.2-1.

Table 6.2.1.9.2-1: Attributes of the TrafficRuleDescriptor

Attribute name	Cardinality	Data type	Description
trafficRuleId	1	String	Identifies the traffic rule.
filterType	1	Enum (inlined)	Definition of filter type: per FLOW or PACKET If it is per FLOW, the filter matches UE->EPC packets and the reverse packets are handled by the same context.
priority	1	Int	Priority of this traffic rule. If traffic rule conflicts, the one with higher priority take precedence.
trafficFilter	1..N	TrafficFilter	The filter used to identify specific flow/packets that need to be handled by the MEC host.
action	1	Enum (inlined)	Identifies the action of the MEC host data plane, when a packet matches the trafficFilter, the example actions includes: DROP, FORWARD_DECAPSULATED, FORWARD_AS_IS, PASSTHROUGH, DUPLICATED_DECAPSULATED, DUPLICATE_AS_IS.
dstInterface	0..2	InterfaceDescriptor	Describes the destination interface information, if the action is FORWARD. Some applications (e.g. inline/tap) require two interfaces, where the first is on the UE side and the second is on the EPC side.

6.2.1.10 Type: TrafficFilter

6.2.1.10.1 Description

The TrafficFilter data type supports the specification of MEC application requirements related to traffic rules.

6.2.1.10.2 Attributes

The attributes of TrafficFilter is shown in table 6.2.1.10.2-1.

Table 6.2.1.10.2-1: Attributes of the TrafficFilter

Attribute name	Cardinality	Data type	Description
srcAddress	0..N	String	An IP address or a range of IP addresses. For IPv4, the IP address could be an IP address plus mask, or an individual IP address, or a range of IP addresses. For IPv6, the IP address could be an IP prefix, or a range of IP prefixes.
dstAddress	0..N	String	A IP address or a range of IP addresses. For IPv4, the IP address could be an IP address plus mask, or an individual IP address, or a range of IP addresses. For IPv6, the IP address could be an IP prefix, or a range of IP prefixes.
srcPort	0..N	String	A port or a range of ports.
dstPort	0..N	String	A port or a range of ports.
protocol	0..N	String	Specify the protocol of the traffic filter.
tag	0..N	String	Used for tag based traffic rule.
srcTunnelAddress	0..N	String	Used for GTP tunnel based traffic rule.
tgtTunnelAddress	0..N	String	Used for GTP tunnel based traffic rule.
srcTunnelPort	0..N	String	Used for GTP tunnel based traffic rule.
dstTunnelPort	0..N	String	Used for GTP tunnel based traffic rule.
qCI	0..1	Int	Used to match all packets that have the same QCI.
dSCP	0..1	Int	Used to match all IPv4 packets that have the same DSCP.
tC	0..1	Int	Used to match all IPv6 packets that have the same TC.

6.2.1.11 Type: InterfaceDescriptor

6.2.1.11.1 Description

The InterfaceDescriptor data type describes an interface of a MEC application.

6.2.1.11.2 Attributes

The attributes of InterfaceDescriptor is shown in table 6.2.1.11.2-1.

Table 6.2.1.11.2-1: Attributes of the InterfaceDescriptor

Attribute name	Cardinality	Data type	Description
interfaceType	1	Enum (inlined)	Type of interface: TUNNEL, MAC, IP, etc.
tunnelInfo	0..1	TunnelInfo	Included only if the destination address type is tunnel.
srcMACAddress	0..1	String	If the interface type is MAC, the source address identifies the MAC address of the interface.
dstMACAddress	0..1	String	If the interface type is MAC, the destination address identifies the MAC address of the destination. Only used for dstInterface.
dstIPAddress	0..1	String	If the interface type is IP, the destination address identifies the IP address of the destination. Only used for dstInterface.

6.2.1.12 Type: TunnelInfo

6.2.1.12.1 Description

The TunnelInfo data type supports the specification of MEC application requirements related to traffic rules.

6.2.1.12.2 Attributes

The attributes of TunnelInfo is shown in table 6.2.1.12.2-1.

Table 6.2.1.12.2-1: Attributes of the TunnelInfo

Attribute name	Cardinality	Data type	Description
tunnelType	1	Enum (inlined)	Type of tunnel: GTP-U, GRE, etc.
tunnelDstAddress	1	String	Destination address of the tunnel.
tunnelSrcAddress	1	String	Source address of the tunnel.
tunnelSpecificData	0..1	Not Specified	Parameters specific to the tunnel.

6.2.1.13 Type: DNSRuleDescriptor

6.2.1.13.1 Description

The DNSRuleDescriptor data type describes DNS rules associated with a MEC application.

6.2.1.13.2 Attributes

The attributes of DNSRuleDescriptor is shown in table 6.2.1.13.2-1.

Table 6.2.1.13.2-1: Attributes of the DNSRuleDescriptor

Attribute name	Cardinality	Data type	Description
dnsRuleId	1	String	Identifies the DNS Rule
domainName	1	String	FQDN of the DNS rule
ipAddressType	1	Enum (inlined)	Specifies the IP address type, value: IP_V6, IP_V4
ipAddress	1	String	IP address given by the DNS rule
ttl	0..1	Int	Time-to-live value

6.2.1.14 Type: LatencyDescriptor

6.2.1.14.1 Description

The LatencyDescriptor data type describes latency requirements for a MEC application.

6.2.1.14.2 Attributes

The attributes of LatencyDescriptor is shown in table 6.2.1.14.2-1.

Table 6.2.1.14.2-1: Attributes of the LatencyDescriptor

Attribute name	Cardinality	Data type	Description
maxLatency	1	Uint32	The value of the maximum latency in nano seconds tolerated by the MEC application. See note.
NOTE: The latency is considered to be the one way end-to-end latency between the client application (e.g. in a device) and the service (i.e. the MEC Application instance).			

6.2.1.15 Type: TerminateAppInstanceOpConfig

6.2.1.15.1 Description

The TerminateAppInstanceOpConfig data type supports the specification of MEC application requirements related to terminate application instance operation configuration.

6.2.1.15.2 Attributes

The attributes of TerminateAppInstanceOpConfig shall follow the definition in clause 7.1.5.7 of ETSI GS NFV-IFA 011 [1].

6.2.1.16 Type: ChangeAppInstanceStateOpConfig

6.2.1.16.1 Description

The ChangeAppInstanceStateOpConfig data type supports the specification of MEC application requirements related to change application instance state operation configuration.

6.2.1.16.2 Attributes

The attributes of ChangeAppInstanceStateOpConfig shall follow the definition in clause 7.1.5.8 of ETSI GS NFV-IFA 011 [1].

6.2.1.17 Type: ServiceDependency

6.2.1.17.1 Description

The ServiceDependency data type supports the specification of requirements of a service-consuming MEC application related to a MEC service.

6.2.1.17.2 Attributes

Attributes of a ServiceDependency are depicted in table 6.2.1.17.2-1.

Table 6.2.1.17.2-1: Attributes of ServiceDependency

Attribute name	Cardinality	Data type	Description
serName	1	String	The name of the service, for example, RNIS, LocationService, etc.
serCategory	0..1	CategoryRef	A Category reference of the service.
version	1	String	The version of the service.
serTransportDependencies	0..N	TransportDependency	Indicates transport and serialization format dependencies of consuming the service. Defaults to REST + JSON if absent. See note.
requestedPermissions	0..N	Not specified	Requested permissions regarding the access of the application to the service. See clause 8.2 of ETSI GS MEC 009 [4]. The format of this attribute is left for the data model design stage.
NOTE: This attribute indicates groups of transport bindings that a service-consuming MEC application supports for the consumption of the MEC service defined by this ServiceDependency structure. If at least one of the indicated groups is supported by the service it may be consumed by the application.			

6.2.1.18 Type: TransportDependency

6.2.1.18.1 Description

The TransportDependency data type supports the specification of requirements of a MEC application related to supported transport bindings (each being a combination of a transport with one or more serializers).

6.2.1.18.2 Attributes

The attributes of a TransportDependency are depicted in table 6.2.1.18.2-1.

Table 6.2.1.18.2-1: Attributes of the TransportDependency

Attribute name	Cardinality	Data type	Description
transport	1	TransportDescriptor	Information about the transport in this transport binding.
serializers	1..N	SerializerTypes	Information about the serializers in this transport binding, as defined in the SerializerTypes type in ETSI GS MEC 011 [i.4]. Support for at least one of the entries is required in conjunction with the transport.
labels	1..N	String	Set of labels that allow to define groups of transport bindings. The mechanism of the grouping is defined below this table.

Each "labels" value identifies a group of transport bindings. In a list of TransportDependency structures, all entries that have a "labels" entry with the same value belong to the same group. Each group indicates an alternative set of transport bindings. At least one group of transport bindings needs to be supported to fulfil the requirements.

EXAMPLE 1: An application requires REST_HTTP transport with JSON.

List of TransportDependency structures:

{transport=REST_HTTP, serializers=[JSON], labels=[A]}

EXAMPLE 2: An application can run with JSON or PROTOBUF3 over a topic-based message bus.

List of TransportDependency structures:

{ transport=MB_TOPIC_BASED, serializers=[JSON, PROTOBUF3], labels=[A])

EXAMPLE 3: An application requires REST transport with JSON or a topic-based message bus with PROTOBUF3.

List of TransportDependency structures:

{transport=REST_HTTP, serializers=[JSON], labels=[A]},

{MB_TOPIC_BASED, serializers=[PROTOBUF3], labels=[B]}

EXAMPLE 4: An application requires both REST transport with JSON and a topic-based message bus with PROTOBUF3.

List of TransportDependency structures:

{transport=REST_HTTP, serializers=[JSON], labels=[A]},

{transport=MB_TOPIC_BASED, serializers=[PROTOBUF3], labels=[A]}

EXAMPLE 5: An application requires both REST transport with JSON and a topic-based message bus with PROTOBUF3 or Websockets with PROTOBUF3.

List of TransportDependency structures:

{transport=REST_HTTP, serializers=[JSON], labels=[A, B]},

{transport=MB_TOPIC_BASED, serializers=[PROTOBUF3], labels=[A]}

{transport=WEBSOCKETS, serializers=[PROTOBUF3], labels=[B]}

6.2.1.19 Type: TransportDescriptor

6.2.1.19.1 Description

The TransportDescriptor data type describes a transport.

6.2.1.19.2 Attributes

The attributes of a TransportDescriptor are depicted in table 6.2.1.19.2-1.

Table 6.2.1.19.2-1: Attributes of the TransportDescriptor

Attribute name	Cardinality	Data type	Description
type	1	TransportTypes	Type of the transport, as defined in the TransportTypes type in ETSI GS MEC 011 [i.4].
protocol	1	String	The name of the protocol used. Shall be set to "HTTP" for a REST API.
version	1	String	The version of the protocol used.
security	1	SecurityInfo	Information about the security used by the transport in ETSI GS MEC 011 [i.4].

6.2.2 Application lifecycle management information model

6.2.2.1 Introduction

This clause defines data structure to be used by application lifecycle management information model.

6.2.2.2 Type: LocationConstraints

6.2.2.2.1 Description

The LocationConstraints data type supports the specification of MEC application requirements related to MEC application deployment location constraints. The location constraints shall be presented as a country code, optionally followed by a civic address based on the format defined by IETF RFC 4776 [2].

6.2.2.2.2 Attributes

The attributes of LocationConstraints are shown in table 6.2.2.2.2-1.

Table 6.2.2.2.2-1: Attributes of the LocationConstraints information element

Attribute name	Data type	Cardinality	Description
countryCode	String	1	The two-letter ISO 3166 [3] country code in capital letters.
civicAddressElement	Structure (inlined)	0..N	Zero or more elements comprising the civic address.
>caType	Integer	1	Describe the content type of caValue. The value of caType shall comply with section 3.4 of IETF RFC 4776 [2].
>caValue	String	1	Content of civic address element corresponding to the caType. The format caValue shall comply with section 3.4 of IETF RFC 4776 [2].

6.2.2.3 Type: CreateAppInstanceRequest

6.2.2.3.1 Description

The data type of CreateAppInstanceRequest represents the parameters for creating a new application instance resource. It is used by the resource of application instances in clause 7.5.1. It shall comply with attributes in clause 6.2.2.3.2.

6.2.2.3.2 Attributes

The attributes of CreateAppInstanceRequest data type shall follow the specification in table 6.2.2.3.2-1.

Table 6.2.2.3.2-1: Attributes of the CreateAppInstanceRequest

Attribute name	Cardinality	Data type	Description
appDId	1	String	The application descriptor identifier. It is managed by the application provider to identify the application descriptor in a globally unique way.
appInstanceName	0..1	String	Human-readable name of the application instance to be created.
appInstanceDescription	0..1	String	Human-readable description of the application instance to be created.

6.2.2.4 Type: AppInstanceInfo

6.2.2.4.1 Description

The data type of AppInstanceInfo represents the parameters of instantiated application instance resources. It is used by the resource of application instances in clause 7.5.1 and the resource of individual application instance in clause 7.5.2.

6.2.2.4.2 Attributes

The attributes of AppInstanceInfo data type are specified in the table 6.2.2.4.2-1.

Table 6.2.2.4.2-1: Attributes of AppInstanceInfo

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of the application instance represented by this data type.
appInstanceName	0..1	String	Name of the application instance.
appInstanceDescription	0..1	String	Human-readable description of the application instance to be created.
appDId	1	String	The application descriptor identifier is managed by the application provider to identify the application descriptor in a globally unique way. It is copied from the AppD of the onboarded application package.
appProvider	1	String	The onboarded application package provider name.
appName	1	String	The onboarded application name.
appSoftVersion	1	String	The application software version.
appDVersion	1	String	Version of the application descriptor.
appPkgId	1	String	Identifier of the onboarded application package.
vimConnectionInfo	0..N	VimConnectionInfo	Information about VIM connections to be used for managing the resources for the application instance.
instantiationState	1	Enum (inlined)	Instantiation state of the application instance: • NOT_INSTANTIATED: the application instance is not instantiated. • INSTANTIATED: the application instance has been instantiated.
instantiatedAppState	0..1	Structure (inlined)	Information specific to an instantiated application. This attribute shall be present if the instantiationState attribute value is INSTANTIATED.
>operationalState	1	Enum (inlined)	Operational state is applicable in the instantiation state INSTANTIATED: • STARTED: the application instance is up and running. • STOPPED: the application instance stops operation.
_links	1	Structure (inlined)	Links to resources related to this resource.
>self	1	LinkType	Self referring URI.
>instantiate	0..1	LinkType	Link to the "instantiate" task resource, if the related operation is possible based on the current status of this application instance resource (i.e. application instance in NOT_INSTANTIATED state).

Attribute name	Cardinality	Data type	Description
>terminate	0..1	LinkType	Link to the "terminate" task resource, if the related operation is possible based on the current status of this application instance resource (i.e. application instance is in INSTANTIATED state).
>operate	0..1	LinkType	Link to the "operate" task resource, if the related operation is supported for this application instance, and is possible based on the current status of this application instance resource (i.e. application instance is in INSTANTIATED state).

6.2.2.5 Type: AppInstanceSubscriptionFilter

6.2.2.5.1 Description

This data type represents subscription filter criteria to match application instances. The AppInstanceSubscriptionFilter shall comply with provisions in clause 6.2.2.5.2.

6.2.2.5.2 Attributes

The attributes of the data type are specified in table 6.2.2.5.2-1.

Table 6.2.2.5.2-1: Attributes of AppInstanceSubscriptionFilter

Attribute name	Cardinality	Data type	Description
appInstSelectorType	1	Enum (Inlined)	0 = void 1 = APP_IDENTITY 2 = APP_NAME 3 = APP_D_ID 4 = APP_FROM_PROVIDER
appInstances	0..N	String	If appInstIdSelector = APP_IDENTITY match existing application instances with an "application instance identifier" listed in this attribute. If appInstIdSelector = APP_NAME match existing application instances with an "application instance name" listed in this attribute. If appInstIdSelector = APP_D_ID match existing application instances, or those created in the future whilst the subscription is active, based on the application descriptors identified by one of the "application descriptor identities" listed in this attribute. If appInstIdSelector = APP_FROM_PROVIDER this attribute shall not be included.
appsFromProviders	0..N	Structure (inlined)	Present only if appInstIdSelector = APP_FROM_PROVIDER. Match existing application instances, or those created in the future whilst the subscription is active, that belong to applications from certain providers.
>appProvider	1	String	Name of the application provider to match.
>appProducts	0..N	Structure (inlined)	If present, match application instances that belong to application products with certain product names, from one particular provider.
>>appName	1	String	Name of the application product to match.
>>versions	0..N	Structure (inlined)	If present, match application instances that belong to application products with certain versions and a certain product name, from one particular provider.
>>>appSoftVersion	1	String	Application software version to match.
>>>appDVersion	0..N	String	If present, match application instances that belong to application products with certain appD versions, a certain software version and a certain product name, from one particular provider.

6.2.2.6 Type: AppLcmOpOccSubscriptionFilter

6.2.2.6.1 Description

This data type represents a subscription filter criteria to match an application LCM operation occurrence. The AppLcmOpOccSubscriptionFilter shall comply with provisions in clause 6.2.2.6.2.

6.2.2.6.2 Attributes

The attributes of the data type are specified in table 6.2.2.6.2-1.

Table 6.2.2.6.2-1: Attributes of AppLcmOpOccSubscriptionFilter

Attribute name	Cardinality	Data type	Description
appInstanceSubscriptionFilter	0..1	AppInstanceSubscriptionFilter	If present, this attribute contains filter criteria that selects one or more application instances on which to receive "LCM operation occurrence" notifications.
notificationTypes	0..N	Enum (inlined)	Match particular notification types. Permitted values: AppLcmOperationOccurrenceNotification.
operationTypes	0..N	Enum (inlined)	Type of the LCM operation represented by this application instance LCM operation occurrence. Permitted values: • INSTANTIATE. • OPERATE. • TERMINATE. Match particular application lifecycle operation types for the notification of AppLcmOpOccNotification. May be present if the "notificationTypes" attribute contains the value "AppLcmOpOccurrenceNotification", and shall be absent otherwise.
operationStates	0..N	Enum (inlined)	Type of the LCM operation state represented by this application instance LCM operation occurrence. Permitted values: • STARTING. • PROCESSING. • COMPLETED. • FAILED. Match particular LCM operation state values as reported in notifications of AppLcmOpOccNotification. May be present if the "notificationTypes" attribute contains the value "AppLcmOpOccNotification", and shall be absent otherwise.

6.2.2.7 Type: InstantiateAppRequest

6.2.2.7.1 Description

This data type represents request parameters of the "Instantiate Application" operation. It shall comply with the provisions in clause 6.2.2.7.2, which aligns with the clause 6.3.1.3.

6.2.2.7.2 Attributes

The attributes of data type are specified in the table 6.2.2.7.2-1.

Table 6.2.2.7.2-1: Attributes of InstantiateAppRequest

Attribute name	Cardinality	Data type	Description
virtualComputeDescriptor	0..1	VirtualComputeDescription	Describes CPU, Memory and acceleration requirements of the virtual machine for the application instance to be created. See note 1.
virtualStorageDescriptor	0..N	VirtualStorageDescriptor	Defines descriptors of virtual storage resources to be used by the application instance to be created. See note 1.
selectedMECHostInfo	1..N	MECHostInformation	Describes the information of selected host for the application instance. See note 2.
locationConstraints	0..1	LocationConstraints	Defines the location constraints for the application instance to be created. See note 3.
vimConnectionInfo	0..N	VimConnectionInfo	Information about VIM connections to be used for managing the resources for the application instance, or refer to external / externally-managed virtual links. This attribute shall only be supported and may be present if application-related resource management in direct mode is applicable. See note 2.
NOTE 1: This attribute may be provided in the InstantiateAppRequest structure to override the same attribute in the AppD. NOTE 2: This field applies to Mm3 reference point only. NOTE 3: This field applies to Mm1 reference point only.			

6.2.2.8 Type: OperateAppRequest

6.2.2.8.1 Description

This data type represents request parameters of the "Operate Application" operation. It shall comply with the provisions in clause 6.2.2.8.2, which aligns with the clause 6.3.1.4.

6.2.2.8.2 Attributes

The attributes of data type are specified in the table 6.2.2.8.2-1.

Table 6.2.2.8.2-1: Attributes of OperateAppRequest

Attribute name	Cardinality	Data type	Description
changeStateTo	1	Enum (inlined)	The desired operational state: • STARTED: the application instance is up and running. • STOPPED: the application instance stops operation.

Attribute name	Cardinality	Data type	Description
stopType	0..1	Enum (inlined)	The stop type: • FORCEFUL: it will stop the application immediately after accepting the request. • GRACEFUL: it will first arrange to take the application instance out of service after accepting the request. Once that operation is successful or once the timer value specified in the "gracefulStopTimeout" attribute expires, it will stop the application. See notes 1 and 3.
gracefulStopTimeout	0..1	Integer	The time interval (in seconds) to wait for the application instance to be taken out of service during graceful stop, before stopping the application. See note 1 and note 2.
NOTE 1: The "stopType" and "gracefulStopTimeout" attributes shall be absent, when the "changeStateTo" attribute is equal to "STARTED". NOTE 2: The "gracefulStopTimeout" attribute shall be present, when the "changeStateTo" is equal to "STOPPED" and the "stopType" attribute is equal to "GRACEFUL". The "gracefulStopTimeout" attribute shall be absent, when the "changeStateTo" attribute is equal to "STOPPED" and the "stopType" attribute is equal to "FORCEFUL". NOTE 3: The request shall be treated as if the "stopType" attribute was set to "FORCEFUL", when the "changeStateTo" attribute is equal to "STOPPED" and the "stopType" attribute is absent.			

6.2.2.9 Type: TerminateAppRequest

6.2.2.9.1 Description

This data type represents request parameters of the "Terminate Application Request" operation. It shall comply with the provisions in clause 6.2.2.9.2, which aligns with the clause 6.3.1.7.

6.2.2.9.2 Attributes

The attributes of data type are specified in the table 6.2.2.9.2-1.

Table 6.2.2.9.2-1: Attributes of TerminateAppRequest

Attribute name	Cardinality	Data type	Description
terminationType	1	Enum (inlined)	Indicates whether forceful or graceful termination is requested. See note. • FORCEFUL: it will shut down the application instance and release the resources immediately after accepting the request. See note. • GRACEFUL: it will first arrange to take the application instance out of service after accepting the request. Once the operation of taking the application instance out of service finishes or once the timer value specified in the "gracefulTerminationTimeout" attribute expires, it will shut down the application instance and release the resources.
gracefulTerminationTimeout	0..1	Integer	This attribute is only applicable in case of graceful termination. It defines the time to wait for the application instance to be taken out of service before shutting down the application and releasing the resources. The unit is seconds. If not given and the "terminationType" attribute is set to "GRACEFUL", it is expected to wait for the successful taking out of service of the application, no matter how long it takes, before shutting down the application and releasing the resources.
NOTE: If the application instance is still in service, requesting forceful termination can adversely impact service.			

6.2.2.10 Type: AppInstSubscriptionInfo

6.2.2.10.1 Description

The data type represents a subscription to notification of application instance operational state change. It shall comply with provisions in clause 6.2.2.10.2.

6.2.2.10.2 Attributes

The attributes of data type are specified in the table 6.2.2.10.2-1.

Table 6.2.2.10.2-1: Attributes of AppInstSubscriptionInfo

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of the subscription to application instance operational state change notification.
subscriptionType	1	String	Shall be set to "AppInstanceStateChange".
notificationType	1	Enum	Subscribed notification type: • NOT_INSTANTIATED: the application instance is not instantiated. • STARTED: the application instance is up and running. • STOPPED: the application instance stops operation.
callbackUri	1	Uri	The URI of the endpoint for the notification to be sent to.
_links	1	Structure (inlined)	Links to resources related to this resource.
>self	1	LinkType	URI of this resource.

6.2.2.11 Type: AppInstNotification

6.2.2.11.1 Description

This data type represents an application instance notification for informing the subscribers about operational state of application instance resources. It shall comply with provisions in clause 6.2.2.11.2.

6.2.2.11.2 Attributes

The attributes of data type are specified in the table 6.2.2.11.2-1.

Table 6.2.2.11.2-1: Attributes of AppInstanceNotification

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "notificationId" attribute of all these notifications shall have the same value.
notificationType	1	Enum (inlined)	Discriminator for the different notification types: • NOT_INSTANTIATED: the application instance is not instantiated. • STARTED: the application instance is up and running. • STOPPED: the application instance stops operation.
subscriptionId	1	String	Identifier of the subscription related to this notification.
timeStamp	1	TimeStamp	Date and time of the notification generation.
appInstanceId	1	String	Identifier of application instance.
appPkgId	1	String	Identifier of the onboarded application package.
appDId	1	String	The application descriptor identifier identifies the application package and the application descriptor in a globally unique way.
_links	1	Structure (inlined)	Links to resources related to this notification.
>subscription	1	LinkType	A link to the related subscription.

6.2.2.12 Type: AppInstSubscriptionRequest

6.2.2.12.1 Description

The data type represents the input parameters of "subscription operation" to notification of application lifecycle management for the operational state change of application instance.

6.2.2.12.2 Attributes

The attributes of the data type are specified in the table 6.2.2.12.2-1.

Table 6.2.2.12.2-1: Attributes of AppInstSubscriptionRequest

Name	Data type	Cardinality	Remarks
subscriptionType	String	1	Shall be set to "AppInstanceStateChange".
callbackUri	Uri	1	The URI of the endpoint for the notification to be sent to.
appInstanceState	Enum	0..1	Only send notifications for application instances that are in one of the states listed in this attribute. If this attribute is absent, match all states. Application states: • NOT_INSTANTIATED: the application instance is not instantiated. • STARTED: the application instance is up and running. • STOPPED: the application instance stops operation.
appInstanceSubscriptionFilter	AppInstanceSubscriptionFilter	0..1	Criteria used to filter application instances for which to send notifications related to this subscription.

6.2.2.13 Type: AppLcmOpOcc

6.2.2.13.1 Description

This data type represents an application lifecycle management operation occurrence. It shall comply with the provisions in clause 6.2.2.13.2.

6.2.2.13.2 Attributes

The attributes of data type are specified in the table 6.2.2.13.2-1.

Table 6.2.2.13.2-1: Attributes of AppInstanceLcmOpOcc

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of the subscription to application LCM operation occurrence notification.
operationState	1	Enum (inlined)	Operation state: • STARTING: the LCM operation starting. • PROCESSING: the LCM operation is currently in execution. • COMPLETED: the LCM operation has been completed. FAILED: The LCM operation has failed.
stateEnteredTime	1	TimeStamp	Date and time when the current state was entered.
startTime	1	TimeStamp	Date and time of the start of the operation.
lcmOperation	1	Enum (inlined)	Type of the actual LCM operation represented by this application instance LCM operation occurrence: • INSTANTIATE. • OPERATE. • TERMINATE. This attribute is associated to the operationParams.

Attribute name	Cardinality	Data type	Description
operationParams	0..1	Object	Input parameters of the LCM operation. This attribute shall be formatted according to the request data type of the related LCM operation. The following mapping between LCM operation and the data type of this attribute shall apply: • INSTANTIATE: InstantiateAppRequest. • OPERATE: OperateAppRequest. • TERMINATE: TerminateAppRequest. This attribute shall be present if this data type is returned in a response to reading an individual resource. See note.
_links	1	Structure (inlined)	Links to resources related to this resource.
>self	1	LinkType	URI of this resource.
>appInstance	1	LinkType	Link to the application instance that the operation applies to.
NOTE: This object contains structured data, and shall comply with the provisions of clause 4 of IETF RFC 8259 [5].			

6.2.2.14 Type: AppLcmOpOccSubscriptionRequest

6.2.2.14.1 Description

This data type represents a subscription request to notification of application life cycle management operation occurrence. It shall comply with provisions in clause 6.2.2.14.2.

6.2.2.14.2 Attributes

The attributes of data type are specified in the table 6.2.2.14.2-1.

Table 6.2.2.14.2-1: Attributes of AppLcmOpOccSubscriptionRequest

Attribute name	Cardinality	Data type	Description
subscriptionType	1	String	Shall be set to "AppLcmOpOccStateChange".
callbackUri	1	Uri	The URI of the endpoint for the notification to be sent to.
appLcmOpOccSubscriptionFilter	0..1	AppLcmOpOccSubscriptionFilter	Subscription filter criteria to match specific application LCM operation occurrences.

6.2.2.15 Type: AppLcmOpOccSubscriptionInfo

6.2.2.15.1 Description

This data type represents a subscription to notifications of application life cycle management operation occurrence. It shall comply with provisions in clause 6.2.2.15.2.

6.2.2.15.2 Attributes

The attributes of data type are specified in the table 6.2.2.15.2-1.

Table 6.2.2.15.2-1: Attributes of AppLcmOpOccSubscriptionInfo

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of this subscription resource.
subscriptionType	1	String	Shall be set to "AppLcmOpOccStateChange".
callbackUri	1	Uri	The URI of the endpoint for the notification to be sent to.
_links	1	Structure (inlined)	Links to resources related to this resource.
>self	1	LinkType	URI of this resource.

6.2.2.16 Type: AppLcmOpOccNotification

6.2.2.16.1 Description

This data type represents a notification related to state changes of an application LCM operation occurrence which informs the subscribers. It shall comply with provisions in clause 6.2.2.16.2.

6.2.2.16.2 Attributes

The attributes of data type are specified in the table 6.2.2.16.2-1.

Table 6.2.2.16.2-1: Attributes of AppLcmOpOccNotification

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "notificationId" attribute of all these notifications shall have the same value.
notificationType	1	Enum (inlined)	Discriminator for the different notification types: • STARTING: the LCM operation starting. • PROCESSING: the LCM operation is currently in execution. • COMPLETED: the LCM operation has been completed. • FAILED: The LCM operation has failed.
subscriptionId	1	String	Identifier of the subscription to this notification.
timeStamp	1	TimeStamp	Date and time of the notification generation.
appLcmOpOccId	1	String	Identifier of application lifecycle management operation occurrence.
appInstanceId	1	String	Identifier of application instance.
_links	1	Structure (inlined)	Links to resources related to this notification.
>appInstance	1	LinkType	Link to the resource representing the application instance to which the notified change applies.
>subscription	1	LinkType	Link to the related subscription.
>appLcmOpOcc	1	LinkType	Link to the application lifecycle management operation occurrence that this notification is related to.

6.2.2.17 Type: MECHostInformation

6.2.2.17.1 Description

The data type represents the parameters of MEC host information.

6.2.2.17.2 Attributes

The attributes of the data type are specified in table 6.2.2.17.2-1.

Table 6.2.2.17.2-1: Attributes of MECHostInformation

Attribute name	Cardinality	Data type	Description
hostName	0..1	String	Human-readable name of MEC host.
hostId	1	KeyValuePairs	User defined MEC host ID.

6.2.2.18 Type: VimConnectionInfo

6.2.2.18.1 Description

The VimConnectionInfo data type specifies the connection information of VIM for managing the resources of the application instance.

6.2.2.18.2 Attributes

The attributes of VimConnectionInfo data type are specified in the table 6.2.2.18.2-1.

Table 6.2.2.18.2-1: Attributes of VimConnectionInfo

Attribute name	Cardinality	Data type	Description
id	1	String	The identifier of the VIM Connection. This identifier is managed by the MEO.
vimId	0..1	String	The identifier of the VIM instance. This identifier is managed by the MEO. Shall be present to address additional information about the VIM if such information has been configured into the MEPM by means outside the scope of the present document, and should be absent otherwise.
vimType	1	String	Discriminator for the different types of the VIM information. The value of this attribute determines the structure of the "interfaceInfo" and "accessInfo" attributes, based on the type of the VIM. The set of permitted values is expected to change over time as new types or versions of VIMs become available.
interfaceInfo	0..1	KeyValuePairs	Information about the interface or interfaces to the VIM, if applicable, such as the URI of an interface endpoint to communicate with the VIM. The applicable keys are dependent on the content of vimType. Alternatively, such information may have been configured into the VNFM and bound to the vimId.
accessInfo	0..1	KeyValuePairs	Authentication credentials for accessing the VIM, and other access-related information such as tenants or infrastructure resource groups (see note). The applicable keys are dependent on the content of vimType. If the VimConnectionInfo structure is part of an HTTP response payload body, sensitive attributes that are children of this attributes (such as passwords) shall not be included. If the VimConnectionInfo structure is part of an HTTP request payload body, sensitive attributes that are children of this attribute (such as passwords) shall be present if they have not been provisioned out of band.
extra	0..1	KeyValuePairs	VIM type specific additional information. The applicable structure, and whether or not this attribute is available, is dependent on the content of vimType.

6.2.3 Application package information model

6.2.3.1 Introduction

This clause defines data structures to be used by the APIs of application package management.

6.2.3.2 Type: CreateAppPkg

6.2.3.2.1 Description

The data type CreateAppPkg represents the parameters for creating a new application package resource. It shall comply with attributes in clause 6.2.3.2.2.

6.2.3.2.2 Attributes

The attributes of CreateAppPkg data type shall follow the specification in table 6.2.3.2.2-1.

Table 6.2.3.2.2-1: Attributes of the CreateAppPkg

Attribute name	Cardinality	Data type	Description
appPkgName	1	String	Name of the application package to be onboarded.
appPkgVersion	1	String	Version of the application package to be onboarded. The appPkgName with appPkgVersion can be used to uniquely identify the application package.
appProvider	0..1	String	The provider's name of the application package to be onboarded.
checksum	1	Checksum	Checksum of the onboarded application package.
userDefinedData	0..1	KeyValuePair	User defined data for the application package.
appPkgPath	1	Uri	Address information of the application package. See note.
NOTE: It is for further study how to convey appPkgPath, and align with ETSI GS NFV-SOL 005 [i.7].			

6.2.3.3 Type: AppPkgInfo

6.2.3.3.1 Description

The data type AppPkgInfo represents the parameters for an application package resource. It shall comply with provisions in clause 6.2.3.3.2.

6.2.3.3.2 Attributes

The attributes of AppPkgInfo data type are specified in the table 6.2.3.3.2-1.

Table 6.2.3.3.2-1: Attributes of AppPkgInfo

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of the application package resource. This identifier is allocated by the MEO.
appDId	1	String	The application descriptor identifier. It is managed by the application provider to identify the application package and the application descriptor in a globally unique way. It is copied from the AppD of the onboarded application package.
appProvider	0..1	String	The provider's name of the onboarded application package.
appName	1	String	Name of the onboarded application.
appSoftwareVersion	1	String	Software version of the application. This is updated when there is any change to the software in the onboarded application package.
appDVersion	1	String	Version of the application descriptor.
checksum	1	Checksum	Checksum of the onboarded application package.
softwareImages	1..N	AppPkgSWImageInfo	Information of application software image in application package. See note 1.
additionalArtifacts	0..N	AppPkgArtifactInfo	Additional information of application package artifacts that are not application software images. See note 2.
onboardingState	1	Enum (inlined)	Onboarding state of application package: • CREATED: The application package resource has been created. • UPLOADING: The associated application package content is being uploaded. • PROCESSING: The associated application package content is being processed, e.g. validation. • ONBOARDED: The associated application package content is successfully onboarded.

Attribute name	Cardinality	Data type	Description
operationalState	1	Enum (inlined)	Operational state of the onboarded application package: • ENABLED: the application package can be used for instantiation of new application instances. • DISABLED: the application package cannot be used for further application instantiation requests.
usageState	1	Enum (inlined)	Usage state of the onboarded instance of the application package: • IN_USE: application instances instantiated from this package exist. • NOT_IN_USE: No application instance instantiated from this package exist.
userDefinedData	0..1	KeyValuePair	User defined data for the application package.
_links	1	Structure (inlined)	Links to resources related to this resource.
>self	1	LinkType	Self referring URI.
>appD	1	LinkType	Link to the appD resource.
>appPkgContent	1	LinkType	Link to the "Onboarded application package content" resource.
NOTE 1: The data type of application software image information data model is related to virtualisation method and needs for further study. NOTE 2: The data type of additional information of application package artefacts is for further study.			

6.2.3.4 Type: AppPkgSubscriptionInfo

6.2.3.4.1 Description

The data type represents a subscription to notification of application package management for the onboarding, or operational state change of application package. It shall comply with provisions in clause 6.2.3.4.2.

6.2.3.4.2 Attributes

The attributes of data type are specified in the table 6.2.3.4.2-1.

Table 6.2.3.4.2-1: Attributes of AppPkgSubscriptionInfo

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of the subscription to application package notification.
subscriptionType	1	String	Subscribed notification type..Permitted values: • "AppPackageOnBoarding": notification of the new onboarded application package. • "AppPacakgeOperationChange": notification of the operational state change of onboarded application package. • "AppPackageDeletion": notification of the application package deletion.
callbackUri	1	Uri	The URI of the endpoint for the notification to be sent to.
_links	1	Structure (inlined)	Links to resources related to this resource.
>self	1	LinkType	URI of this resource.

6.2.3.5 Type: AppPkgSubscriptionLinkList

6.2.3.5.1 Description

The data type represents a subscription link list of notification on application package management. It shall comply with provisions in clause 6.2.3.5.2.

6.2.3.5.2 Attributes

The attributes of data type are specified in the table 6.2.3.5.2-1.

Table 6.2.3.5.2-1: Attributes of AppPkgSubscriptionLinkList

Attribute name	Cardinality	Data type	Description
_links	1	Structure (inlined)	Links to resources related to this resource.
>self	1	LinkType	URI of this resource.
>subscriptions	0..N	Structure (inlined)	A link list to the subscriptions to an application package.
>>href	1	Uri	The URI referring to the subscription.
>>subsctiptionType	1	String	Subscribed notification type. It is same as the notification types: • "AppPackageOnBoarding : notification of the new onboarded application package. • "AppPacakgeOperationChange": notification of the operational state change of onboarded application package. • "AppPackageDeletion": notification of the application package deletion.

6.2.3.6 Type: AppPkgNotification

6.2.3.6.1 Description

This data type represents an application package management notification for informing the subscribers about onboarding application package resources. The notification is triggered when a new application package is onboarded.

It shall comply with provisions in clause 6.2.3.6.2.

6.2.3.6.2 Attributes

The attributes of data type are specified in the table 6.2.3.6.2-1.

Table 6.2.3.6.2-1: Attributes of AppPkgNotification

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of this notification. If a notification is sent multiple times due to multiple subscriptions, the "notificationId" attribute of all these notifications shall have the same value.
notificationType	1	String	Discriminator for the different notification types: • "AppPackageOnBoarded": notification of the new onboarded application package. • "AppPacakgeEnabled": notification of the operational state change of onboarded application package. • "AppPacakgeDisabled": notification of the onboarded application package disabled. • "AppPackageDeleted": notification of the application package deleted.
subscriptionId	1	String	Identifier of the subscription to this notification.
timeStamp	1	TimeStamp	Date and time of the notification generation.
appPkgId	1	String	Identifier of the onboarded application package.
appDId	1	String	The application descriptor identifier identifies the application package and the application descriptor in a globally unique way.
operationalState	1	Enum (inlined)	Operational state of the application package: • ENABLED: the application package can be used for instantiation of new application instances. • DISABLED: the application package cannot be used for further application instantiation requests.
_links	1	Structure (inlined)	Links to resources related to this notification.
>subscription	1	LinkType	A link to the related subscription.

6.2.3.7 Type: AppPkgSubscription

6.2.3.7.1 Description

The data type represents the input parameters of "subscription operation" to notification of application package management for the onboarding, or operational state change of application package.

6.2.3.7.2 Attributes

The attributes of data type are specified in the table 6.2.3.7.2-1.

Table 6.2.3.7.2-1: Attributes of AppPkgSubscription

Name	Data type	Cardinality	Remarks
callbackUri	Uri	1	The URI of the endpoint for the notification to be sent to.
subscriptionType	String	1	Subscribe to notification type: • "AppPackageOnBoarding": notification of the new onboarded application package. • "AppPacakgeOperationChange": notification of the operational state change of onboarded application package. • "AppPackageDeletion": notification of the application package deletion.
appPkgFilter	AppPkgFilter	0..N	The attribute-based filter is to filter application packages on which the query applies.

6.2.3.8 Type: AppPkgInfoModifications

6.2.3.8.1 Description

The data type represents modifications of the "AppPkgInfo" data type that can be requested to perform "application package operation".

6.2.3.8.2 Attributes

The attributes of data type are specified in the table 6.2.3.8.2-1.

Table 6.2.3.8.2-1: Attributes of AppPkgInfoModifications

Name	Data type	Cardinality	Remarks
operationalState	Enum (inlined)	1	New value of the "operationalState" attribute of the "OnboardedAppPkgInfo" structure. Permitted values • DISABLED: to disable the individual application package. • ENABLED: to enable the individual application package.

6.2.4 Granting information model

6.2.4.1 Introduction

This clause defines data types used in the granting resource.

6.2.4.2 Type: GrantRequest

6.2.4.2.1 Description

This type represents a grant request. Refer to clause 9.5.2.2 of ETSI GS NFV-SOL 003 [7].

6.2.4.2.2 Attributes

The attributes of the data type are specified in table 6.2.4.2.2-1.

Table 6.2.4.2.2-1: Attributes of GrantRequest

Attribute name	Cardinality	Data type	Description
appInstanceId	1	String	Identifier of the application instance which this grant request is related to. Shall also be provided for application instances that not yet exist but are planned to exist in the future, i.e. if the grant is requested for Instantiate.
appLcmOpOccId	1	String	The identifier of the application lifecycle management operation occurrence associated to the GrantRequest.
appDId	1	String	Identifier of the AppD that defines the application for which the LCM operation is to be granted.
operation	1	Enum (inlined)	The lifecycle management operation for which granting is requested: • INSTANTIATE. • OPERATE. • TERMINATE. See notes 1 and 2.
addResources	0..N	ResourceDefinition	List of resource definitions in the AppD for resources to be added by the LCM operation which is related to this grant request, with one entry per resource.
tempResources	0..N	ResourceDefinition	List of resource definitions in the AppD for resources to be temporarily instantiated during the runtime of the LCM operation which is related to this grant request. See note 3.
removeResources	0..N	ResourceDefinition	Removed by the LCM operation which is related to this grant request, with one entry per resource.
updateResources	0..N	ResourceDefinition	Provides the definitions of resources to be modified by the LCM operation which is related to this grant request, with one entry per resource.
additionalParams	0..1	KeyValuePairs	MEPM, specific to the application and the LCM operation.
_links	1	Structure (inlined)	Links to resources related to this request.
>appLcmOpOcc	1	LinkType	Related lifecycle management operation occurrence.
>appInstance	1	LinkType	Related application instance.
NOTE 1: Other application LCM operations can be executed by the MEPM without requesting granting. NOTE 2: If the granting request is for Instantiate, addResources shall be present. NOTE 3: The MEO will assume that the MEPM will be responsible to both allocate and release the temporary resource during the runtime of the LCM operation. This means, the resource can be allocated and consumed after the "start" notification for the LCM operation is sent by the MEPM, and the resource will be released before the "result" notification of the application LCM operation is sent by the MEPM.			

6.2.4.3 Type: ResourceDefinition

6.2.4.3.1 Description

This type provides information of an existing or proposed resource used by the application. Refer to clause 9.5.3.2 of ETSI GS NFV-SOL 003 [7].

6.2.4.3.2 Attributes

The attributes of the data type are specified in table 6.2.4.3.2-1.

Table 6.2.4.3.2-1: Attributes of ResourceDefinition

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of this "ResourceDefinition" structure, unique at least within the scope of the "GrantRequest" structure.
type	1	Enum (inlined)	Type of the resource definition referenced. Permitted values: • COMPUTE. • VL. • STORAGE. • LINKPORT.
vduId	0..1	String	Reference to the related VDU in the AppD applicable to this resource. Shall only be present if a VDU is applicable to this resource.
resourceTemplateId	1	String	Reference to a resource template, i.e. VirtualLinkDesc, VirtualComputeDesc, AppExtCpd, VirtualStorageDesc in the AppD.
resource	1	Structure (inlined)	Resource information for an existing resource. Shall be present for resources that are planned to be deleted or modified. Shall be absent otherwise.
>vimConnectionInfo	1	VimConnectionInfo	Specifies the connection information of VIM for the resources of the application instance.
>resourceId	1	String	Identifier of the resource in the scope of the VIM.

6.2.4.4 Type: Grant

6.2.4.4.1 Description

This type represents a grant. Refer to clause 9.5.2.3 of ETSI GS NFV-SOL 003 [7].

6.2.4.4.2 Attributes

The attributes of the data type are specified in table 6.2.4.4.2-1.

Table 6.2.4.4.2-1: Attributes of Grant

Attribute name	Cardinality	Data type	Description
id	1	String	Identifier of the grants.
appInstanceId	1	String	Type of the resource definition referenced. Permitted values: • COMPUTE. • VL. • STORAGE. • LINKPORT.
appLcmOpOccId	1	String	Reference to the related VDU in the AppD applicable to this resource. Shall only be present if a VDU is applicable to this resource.
vimConnections	0..N	VimConnectionInfo	Provides information regarding VIM connections that are approved to be used by the MEPM to allocate resources, and provides parameters of these VIM connections. See note 1.
zones	ZoneInfo	0..N	Identifies resource zones where the resources are approved to be allocated by the MEPM.
zoneGroups	ZoneGroupInfo	0..N	Information about groups of resource zones that are related and that the MEO has chosen to fulfil a zoneGroup constraint in the Grant request.
addResources	0..N	GrantInfo	List of resources that are approved to be added, with one entry per resource.
tempResources	0..N	GrantInfo	List of resources that are approved to be temporarily instantiated during the runtime of the lifecycle operation, with one entry per resource.

Attribute name	Cardinality	Data type	Description
removeResources	0..N	GrantInfo	List of resources that are approved to be removed, with one entry per resource.
updateResources	0..N	GrantInfo	List of resources that are approved to be modified, with one entry per resource.
vimAssets	0..1	Structure (inlined)	Information about assets for the application that are managed by the MEO in the VIM, such as software images. See note 2.
>softwareImages	0..N	VimSoftwareImage	Mappings between software images defined in the AppD and software images managed in the VIM.
extVirtualLinks	ExtVirtualLinkData	0..N	Information about external VLs to connect the application instance to. See note 3.
additionalParams	0..1	KeyValuePairs	MEPM, specific to the application and the LCM operation.
_links	1	Structure (inlined)	Links to resources related to this request.
>appLcmOpOcc	1	LinkType	Related lifecycle management operation occurrence.
>appInstance	1	LinkType	Related application instance.
NOTE 1: This interface allows to signal the use of multiple VIMs per application. However, due to the partial support of this feature in the present release, it is recommended in the present document that the number of entries in the "vims" attribute in the Grant is not greater than 1. NOTE 2: The further condition will be defined by ETSI GS NFV-SOL 003 [7]. NOTE 3: External VLs can be passed in the application lifecycle management operation requests such as Instantiate, and/or in the grant response. The MEO may choose to override in the grant response external VL instances that have been passed previously in the associated application lifecycle management request, if the lifecycle management request has originated from the MEO itself.			

6.2.4.5 Type: GrantInfo

6.2.4.5.1 Description

This type contains information about a Compute, storage or network resource whose addition/update/deletion was granted. It shall comply with the provisions defined in table 6.2.4.5.2-1. Refer to clause 9.5.3.3 of ETSI GS NFV-SOL 003 [7].

6.2.4.5.2 Attributes

The attributes of the data type are specified in table 6.2.4.5.2-1.

Table 6.2.4.5.2-1: Definition of the GrantInfo data type

Attribute name	Data type	Cardinality	Description
resourceDefinitionId	String	1	Identifier of the related "ResourceDefinition" structure from the related "GrantRequest" structure.
vimConnectionId	String	0..1	Identifier of the VIM connection to be used to manage this resource. Shall be present for new resources, and shall be absent for resources that have already been allocated.
zoneId	String	0..1	Reference to the identifier of the "ZoneInfo" structure in the "Grant" structure defining the resource zone into which this resource is to be placed. Shall be present for new resources if the zones concept is applicable to them (typically, Compute resources), and shall be absent for resources that have already been allocated.
resourceGroupId	String	0..1	Identifier of the "infrastructure resource group", logical grouping of virtual resources assigned to a tenant within an Infrastructure Domain, to be provided when allocating the resource. If the VIM connection referenced by "vimConnectionId" applies to multiple infrastructure resource groups, this attribute shall be present for new resources. If the VIM connection referenced by "vimConnectionId" applies to a single infrastructure resource group, this attribute may be present for new resources. This attribute shall be absent for resources that have already been allocated.

6.2.4.6 Type: ZoneInfo

6.2.4.6.1 Description

This type provides information regarding a resource zone. Refer to clause 9.5.3.4 of ETSI GS NFV-SOL 003 [7].

6.2.4.6.2 Attributes

The attributes of the data type are specified in table 6.2.4.6.2-1.

Table 6.2.4.6.2-1: Definition of the ZoneInfo data type

Attribute name	Data type	Cardinality	Description
id	String	1	The identifier of this ZoneInfo instance, for the purpose of referencing it from other structures in the "Grant" structure.
zoneId	String	1	The identifier of the resource zone, as managed by the resource management layer (typically, the VIM).
vimConnectionId	String	0..1	Identifier of the connection to the VIM that manages the resource zone. The applicable "VimConnectionInfo" structure, which is referenced by vimConnectionId, can be obtained from the " vimConnectionInfo" attribute of the "App Instance" structure.

6.2.4.7 Type: ZoneGroupInfo

6.2.4.7.1 Description

This type provides information regarding a resource zone group. A resource zone group is a group of one or more related resource zones which can be used in resource placement constraints. To fulfil such constraint, the MEO may decide to place a resource into any zone that belongs to a particular group. Refer to clause 9.5.3.5 of ETSI GS NFV-SOL 003 [7].

NOTE: A resource zone group can be used to support overflow from one resource zone into another, in case a particular deployment supports only non-elastic resource zones.

6.2.4.7.2 Attributes

The attributes of the data type are specified in table 6.2.4.7.2-1.

Table 6.2.4.7.2-1: Definition of the ZoneGroupInfo data type

Attribute name	Data type	Cardinality	Description
zoneId	String	1..N	References of identifiers of "ZoneInfo" structures, each of which provides information about a resource zone that belongs to this group.

6.2.4.8 Type: ExtVirtualLinkData

6.2.4.8.1 Description

This type represents an external VL. Refer to clause 4.4.1.11 of ETSI GS NFV-SOL 003 [7].

6.2.4.8.2 Attributes

It shall comply with the provisions defined in table 6.2.4.8.2-1.

Table 6.2.4.8.2-1: Definition of the ExtVirtualLinkData data type

Attribute name	Data type	Cardinality	Description
id	String	1	The identifier of the external VL instance. The identifier is assigned by the MEC entity that manages this VL instance.
vimConnectionId	String	0..1	Identifier of the VIM connection to manage this resource.
resourceId	String	1	The identifier of the resource in the scope of the VIM.
extCps	AppExtCpData	1..N	External CPs of the application instance to be connected to this external VL.
extLinkPorts	ExtLinkPortData	0..N	Externally provided link ports to be used to connect external connection points to this external VL. If this attribute is not present, the MEPM shall create the link ports on the external VL.

6.2.4.9 Type: ExtLinkPortData

6.2.4.9.1 Description

This type represents an externally provided link port to be used to connect an external connection point to an external VL. Refer to clause 5.5.3.9a of ETSI GS NFV-SOL 003 [7].

6.2.4.9.2 Attributes

It shall comply with the provisions defined in table 6.2.4.9.2-1.

Table 6.2.4.9.2-1: Definition of the ExtLinkPortData data type

Attribute name	Data type	Cardinality	Description
id	String	1	Identifier of this link port as provided by the entity that has created the link port.
resourceHandle	ResourceHandle	1	Reference to the virtualised resource realizing this link port.

6.2.4.10 Type: ResourceHandle

6.2.4.10.1 Description

This type represents the information that allows addressing a virtualised resource that is used by an application instance. Information about the resource is available from the VIM. Refer to clause 4.4.1.7 of ETSI GS NFV-SOL 003 [7].

6.2.4.10.2 Attributes

It shall comply with the provisions defined in table 6.2.4.10.2-1.

Table 6.2.4.10.2-1: Definition of the ResourceHandle data type

Attribute name	Data type	Cardinality	Description
vimConnectionId	String	0..1	Identifier of the VIM connection to manage the resource. The applicable "VimConnectionInfo" structure, which is referenced by vimConnectionId, can be obtained from the "vimConnectionInfo" attribute of the "AppInstance" structure.
resourceId	String	1	Identifier of the resource in the scope of the VIM.
vimLevelResourceType	String	0..1	Type of the resource in the scope of the VIM. See note.
NOTE: The value set of the "vimLevelResourceType" attribute is within the scope of the VIM and can be used as information that complements the ResourceHandle. This value set is different from the value set of the "type" attribute in the ResourceDefinition.			

6.2.4.11 Type: VimSoftwareImage

6.2.4.11.1 Description

This type contains a mapping between a software image definition in the AppD and the corresponding software image managed by the MEO in the VIM which is needed during compute resource instantiation. Refer to clause 9.5.3.10 of ETSI GS NFV-SOL 003 [7].

6.2.4.11.2 Attributes

It shall comply with the provisions defined in table 6.2.4.11.2-1.

Table 6.2.4.11.2-1: Definition of the VimSoftwareImage data type

Attribute name	Data type	Cardinality	Description
vimConnectionId	String	0..1	Identifier of the VIM connection to access the software image referenced in this structure.
appDSoftwareImageId	String	1	Identifier which references the software image descriptor in the AppD.
vimSoftwareImageId	String	1	Identifier of the software image in the resource management layer (i.e. VIM).

6.2.4.12 Type: AppExtCpData

6.2.4.12.1 Description

This type represents configuration information for external CPs created from a CPD. Refer to clause 4.4.1.10 of ETSI GS NFV-SOL 003 [7].

6.2.4.12.2 Attributes

It shall comply with the provisions defined in table 6.2.4.12.2-1.

Table 6.2.4.12.2-1: Definition of the AppExtCpData data type

Attribute name	Data type	Cardinality	Description
cpdId	String	1	The identifier of the CPD in the AppD.
cpConfig	AppExtCpConfig	1..N	List of instance data that need to be configured on the CP instances created from the respective CPD.

6.2.4.13 Type: AppExtCpConfig

6.2.4.13.1 Description

This type represents an externally provided link port or network address information per instance of an external connection point. In case a link port is provided, the MEPM shall use that link port when connecting the external CP to the external VL. In case a link port is not provided, the MEPM shall create a link port on the external VL, and use that link port to connect the external CP to the external VL. Refer to clause 4.4.1.10a of ETSI GS NFV-SOL 003 [7].

6.2.4.13.2 Attributes

This type shall comply with the provisions defined in table 6.2.4.13.2-1.

Table 6.2.4.13.2-1: Definition of the AppExtCpConfig data type

Attribute name	Data type	Cardinality	Description
cpInstanceId	String	0..1	Identifier of the external CP instance to which this set of configuration parameters is requested to be applied. Shall be present if this instance has already been created.
linkPortId	String	0..1	Identifier of a pre-configured link port to which the external CP will be associated. See note.
cpProtocolData	CpProtocolData	0..N	Parameters for configuring the network protocols on the link port that connects the CP to a VL. See note.
NOTE: The following conditions apply to the attributes "linkPortId" and " cpProtocolData": 1) The "linkPortId" and "cpProtocolData" attributes shall both be absent for the deletion of an existing external CP instance addressed by cpInstanceId. 2) At least one of these attributes shall be present for a to-be-created external CP instance or an existing external CP instance. 3) If the "linkPortId" attribute is absent, the MEPM shall create a link port. 4) If the "cpProtocolData" attribute is absent, the "linkPortId" attribute shall be provided referencing a pre-created link port, and the MEPM can use means outside the scope of the present document to obtain the pre-configured address information for the connection point from the resource representing the link port. 5) If both "cpProtocolData" and "linkportId" are provided, the API consumer shall ensure that the cpProtocolData can be used with the pre-created link port referenced by "linkPortId".			

6.2.4.14 Type: CpProtocolData

6.2.4.14.1 Description

This type represents network protocol data. Refer to clause 4.4.1.10b of ETSI GS NFV-SOL 003 [7].

6.2.4.14.2 Attributes

This type shall comply with the provisions defined in table 6.2.4.14.2-1.

Table 6.2.4.14.2-1: Definition of the CpProtocolData data type

Attribute name	Data type	Cardinality	Description
layerProtocol	Enum (inlined)	1	Identifier of layer(s) and protocol(s). Permitted values: IP_OVER_ETHERNET. See note.
ipOverEthernet	IpOverEthernetAddressData	0..1	Network address data for IP over Ethernet to assign to the extCP instance. Shall be present if layerProtocol is equal to "IP_OVER_ETHERNET", and shall be absent otherwise.
NOTE: This attribute allows to signal the addition of further types of layer and protocol in future versions of the present document in a backwards-compatible way. In the current version of the present document, only IP over Ethernet is supported.			

6.2.4.15 Type: IpOverEthernetAddressData

6.2.4.15.1 Description

This type represents network address data for IP over Ethernet. Refer to clause 4.4.1.10c of ETSI GS NFV-SOL 003 [7].

6.2.4.15.2 Attributes

It shall comply with the provisions defined in table 6.2.4.15.2-1.

Table 6.2.4.15.2-1: Definition of the IpOverEthernetAddressData data type

Attribute name	Data type	Cardinality	Description
macAddress	String	0..1	MAC address. If this attribute is not present, it shall be chosen by the VIM. See note 1.
ipAddresses	Structure (inlined)	0..N	List of IP addresses to assign to the CP instance. Each entry represents IP address data for fixed or dynamic IP address assignment per subnet. If this attribute is not present, no IP address shall be assigned. See note 1.
>type	Enum (inlined)	1	The type of the IP addresses. Permitted values: IPV4, IPV6.
>fixedAddresses	String	0..N	Fixed addresses to assign (from the subnet defined by "subnetId" if provided). See note 2.
>numDynamicAddresses	Integer	0..1	Number of dynamic addresses to assign (from the subnet defined by "subnetId" if provided). See note 2.
>addressRange	Structure (inlined)	0..1	An IP address range to be used, e.g. in case of egress connections. In case this attribute is present, IP addresses from the range will be used. See note 2.
>>minAddress	String	1	Lowest IP address belonging to the range.
>>maxAddress	String	1	Highest IP address belonging to the range.
>subnetId	String	0..1	Subnet defined by the identifier of the subnet resource in the VIM. In case this attribute is present, IP addresses from that subnet will be assigned; otherwise, IP addresses not bound to a subnet will be assigned.
NOTE 1: At least one of "macAddress" or "ipAddresses" shall be present. NOTE 2: Exactly one of "fixedAddresses", "numDynamicAddresses" or "ipAddressRange" shall be present.			

6.2.5 Common information model

6.2.5.1 Introduction

This clause defines common data structures used by other information models.

6.2.5.2 Type: LinkType

6.2.5.2.1 Description

This data type represents a type of link.

6.2.5.2.2 Attributes

The attributes of LinkType are specified in the table 6.2.5.2.2-1.

Table 6.2.5.2.2-1: Attributes of LinkType

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

6.2.5.3 Type: KeyValuePairs

6.2.5.3.1 Description

This data type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 8259 [5]. In the following example, a list of key-value pairs with four keys ("aString", "aNumber", "anArray" and"anObject") is provided to illustrate that the values associated with different keys can be of different type.

EXAMPLE:

```
{
   "aString" : "ETSI ISG MEC",
   "aNumber" : 0.01,
   "anArray" : [1,2,3],
   "anObject" : {"organization" : "ETSI", "ISG" : "MEC"}
}
```

6.2.5.4 Type: TimeStamp

6.2.5.4.1 Description

This data type represents the time stamp as Unix-time since January 1, 1970, 00:00:00 UTC.

The TimeStamp shall comply with provisions in clause 6.2.5.4.2.

6.2.5.4.2 Attributes

The attributes of data type are specified in the table 6.2.5.4.2-1.

Table 6.2.5.4.2-1: Attributes of TimeStamp

Attribute name	Cardinality	Data type	Description
seconds	1	Uint32	The seconds part of the Time. Time is defined as Unix-time since January 1, 1970, 00:00:00 UTC.
nanoSeconds	1	Uint32	The nanoseconds part of the Time. Time is defined as Unix-time since January 1, 1970, 00:00:00 UTC.

6.2.5.5 Type: SubscriptionLinkList

6.2.5.5.1 Description

The data type represents a subscription link list of notification on application lifecycle management. It shall comply with provisions in clause 6.2.5.5.2.

6.2.5.5.2 Attributes

The attributes of data type are specified in the table 6.2.5.5.2-1.

Table 6.2.5.5.2-1: Attributes of SubscriptionLinkList

Attribute name	Cardinality	Data type	Description
_links	1	Structure (inlined)	List of hyperlinks related to the resource.
>self	1	LinkType	URI of this resource.
>subscriptions	0..N	Structure (inlined)	A link list to the subscriptions.
>>href	1	Uri	The URI referring to the subscription.
>>subscriptionType	1	String	Permitted values: • "AppInstanceStateChange". • "AppLcmOpOccStateChange".

6.3 Interfaces

6.3.1 Application lifecycle management interface

6.3.1.1 Description

This interface allows the OSS to invoke lifecycle management operations towards the MEO or allows the MEO to invoke lifecycle management operations towards the MEPM.

The following operations are defined:

- Create application instance identifier.
- Application instantiation.
- Application instance terminate.
- Delete application instance identifier.
- Query application instance information.
- Change application instance state.
- Query application lifecycle operation Status.
- Subscribe to notifications relating to application lifecycle management.

An identifier (i.e. lifecycleOperationOccurrenceId) is generated for each application lifecycle operation occurrence, except for query application instance information, create application instance identifier, delete application instance identifier, query application lifecycle operation status and subscribe to notifications relating to application lifecycle management.

6.3.1.2 Create application instance identifier operation

6.3.1.2.1 Description

This operation creates an application instance identifier, and an associated instance of an AppInstanceInfo, identified by that identifier, in the NOT_INSTANTIATED state without instantiating the application or doing any additional lifecycle operation(s). It allows returning right away an application instance identifier that may be used in subsequent lifecycle operations, like the application instantiation operation.

Table 6.3.1.2.1-1 lists the information flow exchanged between the MEPM and the MEO and between MEO and OSS.

Table 6.3.1.2.1-1: Create application instance identifier operation

Message	Requirement	Direction
CreateAppInstanceIdentifierRequest	Mandatory	MEO → MEPM, OSS → MEO
CreateAppInstanceIdentifierResponse	Mandatory	MEPM → MEO, MEO → OSS

6.3.1.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.1.2.2-1.

Table 6.3.1.2.2-1: Create application instance identifier operation input parameters

Attribute name	Cardinality	Data type	Description
appDId	1	String	Identifier that identifies the application package that will be instantiated. The application package is identified by the appDId of the application descriptor included in the package.
appInstanceName	0..1	String	Human-readable name of the application instance to be created.
appInstanceDescription	0..1	String	Human-readable description of the application instance to be created.

6.3.1.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.1.2.3-1.

Table 6.3.1.2.3-1: Create application instance identifier operation output parameters

Attribute name	Cardinality	Data type	Description
appInstanceId	1	String	The application instance identifier just created.

6.3.1.2.4 Operation results

In case of success, an instance of an AppInstanceInfo, in the NOT_INSTANTIATED state, has been created and may be used in subsequent lifecycle operations and the corresponding AppInstanceIdentifierCreationNotification has been sent. In case of failure, appropriate error information is returned.

6.3.1.3 Application instantiation operation

6.3.1.3.1 Definition

This operation instantiates a MEC application instance.

Table 6.3.1.3.1-1 lists the information flow for application instantiation.

Table 6.3.1.3.1-1: InstantiateApp operation

Message	Requirement	Direction
InstantiateAppRequest	Mandatory	OSS → MEO, MEO → MEPM
InstantiateAppResponse	Mandatory	MEO → OSS, MEPM → MEO

6.3.1.3.2 Input parameters

The input parameters for this operation is shown in table 6.3.1.3.2-1.

Table 6.3.1.3.2-1: InstantiateApp operation

Attribute name	Cardinality	Data type	Description
appInstanceId	1	String	Identifier of the application instance created by "Create application instance identifier" operation.
virtualComputeDescriptor	0..1	VirtualComputeDescription	Describes CPU, Memory and acceleration requirements of the virtual machine for the MEC application instance to be created. See note 1.
virtualStorageDescriptor	0..N	VirtualStorageDescriptor	Defines descriptors of virtual storage resources to be used by the MEC application instance to be created. See note 1.
selectedMECHostInfo	1..N	MECHostInformation	Describes the information of selected MEC host for the application instance. See notes 2 and 3.

Attribute name	Cardinality	Data type	Description
locationConstraints	0..1	LocationConstraints	Defines the location constraints for the MEC application instance to be created. See note 4.
NOTE 1: The same field of AppD will be override by the value of the field in this table. NOTE 2: This field applies to Mm3 reference point only. NOTE 3: This data type is not specified in the present document. NOTE 4: This field applies to Mm1 reference point only.			

6.3.1.3.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.1.3.3-1.

Table 6.3.1.3.3-1: InstantiateApp operation output parameters

Attribute name	Cardinality	Data type	Description
lifecycleOperationOccurrenceId	1	String	The identifier of the application lifecycle operation occurrence.

6.3.1.3.4 Operation results

In case of success, the MEC application has been instantiated, initially configured, and Lifecycle Change Notifications have been sent accordingly. In case of failure, appropriate error information is returned in lifecycle change notification.

The responder shall first return the lifecycleOperationOccurrenceId and second send the "start" Lifecycle Change Notification before additional notifications or messages as part of this operation are issued, or operations towards the VIM are invoked.

On successful as well as unsuccessful completion of the operation, the responder shall send the "result" Lifecycle Change Notification.

6.3.1.4 Change application instance operational state operation

6.3.1.4.1 Description

This operation enables requesting to change the state of a MEC application instance, including starting and stopping the application instance.

NOTE 1: These operations are complementary to instantiating and terminating a MEC application instance.

NOTE 2: In the present document, only starting and stopping the MEC application instance(s) are supported. Extension of this operation to support other MEC application state changes is left for future specification.

A MEC application instance may be in the following states:

- STARTED: the MEC application instance is up and running.
- STOPPED: the MEC application instance has been shut down.

In the state STOPPED, the virtualised machine, where the MEC application instance run, are shut down but not terminated. In addition, if the workflow requires a graceful stop, as part of this process, the MEC platform will interact with the MEC application instance to gracefully stop the MEC application. Once a MEC application is instantiated, i.e. all instantiation steps have been completed, the MEC application instance is in the state STARTED.

Figure 6.3.1.4.1-1 illustrates the application instance operational state diagram. The desired change of state is indicated as an input in the OperateAppInstanceRequest message.

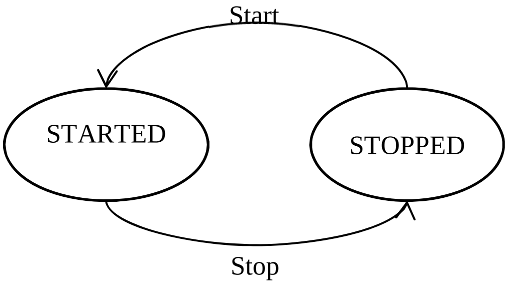

Figure 6.3.1.4.1-1: Change application instance operational state diagram

It depends on the MEC application capabilities, and is declared in the AppD, whether this operation is supported for a particular MEC application.

Table 6.3.1.4.1-1 lists the information flow exchanged between the initiator and the responder.

Table 6.3.1.4.1-1: Change application instance state operation

Message	Requirement	Direction
OperateAppInstanceRequest	Mandatory	OSS → MEO MEO → MEPM
OperateAppInstanceResponse	Mandatory	MEO → OSS MEPM → MEO

6.3.1.4.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.1.4.2-1.

Table 6.3.1.4.2-1: Change application instance state operation input parameters

Attribute name	Cardinality	Data type	Description
appInstanceId	1	String	Identifier of the MEC application instance.
changeStateTo	1	Enum (inlined)	The desired state to change the VNF to. Permitted values are: STARTED, STOPPED.
stopType	0..1	Enum (inlined)	Signals whether forceful or graceful stop is requested. Allowed values are: FORCEFUL and GRACEFUL. In case of FORCEFUL stop, the MEC application is stopped immediately. Note that if the MEC application is still in service, this may adversely impact network service, and therefore, operator policies apply to determine if FORCEFUL stop is allowed in the particular situation. In case of GRACEFUL stop, the MEC system gives time to the MEC application for application level stop (e.g. via Mp1 interaction). Once this was successful, or after a timeout, the MEC system stops the MEC application. If the MEC application does not support Mp1, the stopType shall be set to FORCEFUL.

Attribute name	Cardinality	Data type	Description
gracefulStopTimeout	0..1	Int	The time interval to wait for the application instance to stop service during graceful stop, before stopping the application instance. If not given, it is expected that the MEC system waits for the successful application level stop, no matter how long it takes, before stopping the MEC application (see note). Minimum timeout or timeout range are specified by the application vendor defined in the AppD. Not relevant in case of forceful stop.
NOTE: This implies that no application instance stop will be attempted if application level stopping fails or hangs.			

6.3.1.4.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.1.4.3-1.

Table 6.3.1.4.3-1: Change application instance state operation output parameters

Attribute name	Cardinality	Data type	Description
lifecycleOperationOccurrenceId	1	String	The identifier of the MEC application lifecycle operation occurrence.

6.3.1.4.4 Operation results

In case of success, the state of the MEC application instance has been changed. In case of failure, appropriate error information is provided in the "result" Lifecycle Change Notification.

The producer shall first return the lifecycleOperationOccurrenceId and second send the "start" Lifecycle Change Notification before additional notifications or messages as part of this operation are issued, or operations towards the VIM are invoked.

On successful as well as unsuccessful completion of the operation, the MEPM shall send the "result" Lifecycle Change Notification.

6.3.1.5 Query application instance information operation

6.3.1.5.1 Description

This operation provides information about application instances. The applicable application instances may be chosen based on filtering criteria, and the information may be restricted to selected attributes.

Table 6.3.1.5.1-1 lists the information flow exchanged between the OSS and the MEO, or the MEO and MEPM.

Table 6.3.1.5.1-1: Query application instance information operation

Message	Requirement	Direction
QueryAppInstanceInfoRequest	Mandatory	OSS → MEO, MEO → MEPM
QueryAppInstanceInfoResponse	Mandatory	MEO → OSS, MEPM → MEO

6.3.1.5.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.1.5.2-1.

Table 6.3.1.5.2-1: Query application instance information operation input parameters

Attribute name	Cardinality	Data type	Description
filter	1	Filter	Filter to select the application instance(s) about which information is queried. See note.
attributeSelector	0..N	String	Provides a list of attribute names. If present, only these attributes will be returned for the application instance(s) matching the filter. If absent, the complete information will be returned for the application instance(s) matching the filter.
NOTE: See Table 7.4.1.3.2-1 for the attribute-based filter and selector.			

6.3.1.5.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.1.5.3-1.

Table 6.3.1.5.3-1: Query application instance information operation output parameters

Attribute name	Cardinality	Data type	Description
appInstanceInfo	0..N	AppInstanceInfo	The information about the selected application instance(s) that are returned. If attributeSelector is present, only the attributes listed in attributeSelector will be returned for the selected application instance(s). See note.
NOTE: The lower cardinality is 0 since there may be no matches to the provided filter.			

6.3.1.5.4 Operation results

In case of success, information related to the application instances that match the filter is returned. In case of failure, appropriate error information is returned.

6.3.1.6 Query application lifecycle operation status

6.3.1.6.1 Description

This operation provides the status of an application lifecycle management operation.

Table 6.3.1.6.1-1 lists the information flow exchanged between the OSS and the MEO or the MEO and the MEPM.

Table 6.3.1.6.1-1: Query application lifecycle operation status operation

Message	Requirement	Direction
QueryAppLcmOperationStatusRequest	Mandatory	OSS → MEO, MEO → MEPM
QueryAppLcmOperationStatusResponse	Mandatory	MEO → OSS, MEPM → MEO

6.3.1.6.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.1.6.2-1.

Table 6.3.1.6.2-1: Query application lifecycle operation status operation input parameters

Attribute name	Cardinality	Data type	Description
lifecycleOperationOccurrenceId	1	String	Identifier of the application lifecycle operation occurrence.

6.3.1.6.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.1.6.3-1.

Table 6.3.1.6.3-1: Query application lifecycle operation status operation output parameters

Attribute name	Cardinality	Data type	Description
operationStatus	1	Enum (inlined)	Indicates the operation status (which includes, for example: PROCESSING, SUCCESSFULLY_DONE, FAILED and operation-specific states).

6.3.1.6.4 Operation results

In success of the operation, the status of the queried operation will be returned. In case of failure, appropriate error code will be returned.

6.3.1.7 Application instance terminate operation

6.3.1.7.1 Description

This operation terminates a MEC application instance.

A MEC application instance may be terminated gracefully or forcefully. Graceful termination means that the MEC Platform Manager gives time to the MEC application for application level termination, and after the MEC application has terminated in application level, the MEC system releases the resources used by the MEC application. Forceful termination means that the MEC Platform Manager immediately shuts down the MEC application and releases the resources. A time interval is specified for graceful termination, after the timer specified by the time interval expires, the MEC Platform Manager will shut down the MEC application and release the resources. The graceful termination requires that the MEC application supports Mp1 reference point.

Table 6.3.1.7.1-1 lists the information flow exchanged between the initiator and the responder.

Table 6.3.1.7.1-1: TerminateMEApp operation

Message	Requirement	Direction
TerminateAppInsRequest	Mandatory	OSS → MEO, MEO → MEPM
TerminateAppInsResponse	Mandatory	MEO → OSS, MEPM → MEO

6.3.1.7.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.1.7.2-1.

Table 6.3.1.7.2-1: TerminateMEApp operation input parameters

Attribute name	Cardinality	Data type	Description
appInstanceId	1..N	String	Identifier of the MEC application instance to be terminated.

Attribute name	Cardinality	Data type	Description
terminationType	1..N	Enum (inlined)	Signals whether FORCEFUL or GRACEFUL termination is requested. In case of FORCEFUL termination, the MEC application is shut down immediately, and resources are released. Note that if the MEC application is still in service, this may adversely impact user experience. In case of GRACEFUL termination, the MEC system gives time to the MEC application for application level termination (e.g. via Mp1 interaction). Once this was successful, or after a timeout, the MEC system shuts down the MEC application and releases the resources. If the MEC application does not support Mp1, the terminationType shall be set to FORCEFUL termination.
gracefulTerminationTimeout	0..N	Int	The time interval given to MEC application for application level termination during graceful termination, before shutting down the MEC application and releasing the resources. If not given, it is expected that the MEC system waits for the successful application level termination, no matter how long it takes, before shutting down the MEC application and releasing the resources (see note). Minimum timeout or timeout range are specified by the application vendor defined in the AppD. Not relevant in case of forceful termination.
NOTE: This implies that no MEC application shutdown and resource release will be attempted if taking the MEC application out of service fails or hangs.			

6.3.1.7.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.1.7.3-1.

Table 6.3.1.7.3-1: Change application instance state operation output parameters

Attribute name	Cardinality	Data type	Description
lifecycleOperationOccurrenceId	1	String	The identifier of the MEC application lifecycle operation occurrence.

6.3.1.7.4 Operation results

In case of success, the MEC application instance has been terminated, resources used by the MEC application instance have been released. In case of failure, appropriate error information is returned.

6.3.1.8 Delete application instance identifier operation

6.3.1.8.1 Description

This operation deletes an application instance identifier and the associated instance of an AppInstanceInfo in the NOT_INSTANTIATED state.

Table 6.3.1.8.1-1 lists the information flow exchanged between MEO and MEPM, and between OSS and MEO.

Table 6.3.1.8.1-1: Delete application instance Identifier operation

Message	Requirement	Direction
DeleteAppInstanceIdentifierRequest	Mandatory	MEO → MEPM, OSS → MEO
DeleteAppInstanceIdentifierResponse	Mandatory	MEPM → MEO, MEO→ OSS

6.3.1.8.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.1.8.2-1.

Table 6.3.1.8.2-1: Delete application instance Identifier operation input parameters

Attribute name	Cardinality	Data type	Description
appInstanceId	1	String	Application instance identifier to be deleted.

6.3.1.8.3 Output parameters

No output parameter.

6.3.1.8.4 Operation results

In case of success, the application instance identifier and the associated instance of the AppInstanceInfo has been deleted and is no longer used; and the corresponding AppInstanceIdentifierDeletionNotification has been sent. If the application instance was not terminated (i.e. the application instance is in INSTANTIATED state), the operation shall be rejected.

In case of failure, appropriate error information is returned.

6.3.1.9 Subscribe to application lifecycle management notifications

6.3.1.9.1 Description

This operation allows a subscriber to subscribe to notifications relating to MEC application lifecycle management, including notification of operational state changes, as well as notification on the creation/deletion of an application instance identifier, with its associated application instance.

6.3.1.9.2 Subscribe

6.3.1.9.2.1 Description

The subscriber subscribes with a filter to the notifications related to MEC application lifecycle management operational state changes, as well as creation/deletion of MEC application instance identifiers and the associated application instances.

Table 6.3.1.9.2.1-1 lists the information flow exchanged between OSS and MEO, or MEPM and MEO.

Table 6.3.1.9.2.1-1: Subscribe operation

Message	Requirement	Direction
SubscribeRequest	Mandatory	OSS →MEO, or MEO → MEPM
SubscribeResponse	Mandatory	MEO →OSS, or MEPM → MEO

6.3.1.9.2.2 Input parameters

The input parameters of a subscribe request shall follow the information in table 6.3.1.9.2.2-1.

Table 6.3.1.9.2.2-1: Subscribe operation input parameters

Attribute name	Cardinality	Data type	Description
filter	1	AppInstSubscriptionRequest	Input filter for selecting e.g. the application instances of interest and the specific types of changes. See note.
NOTE: When subscribing for notifications regarding the creation of application instance identifiers and the associated application instance information object instances, selecting the application instances in the filter is not possible.			

6.3.1.9.2.3 Output parameters

The output parameters returned in the response to a subscribe request shall follow the indications in table 6.3.1.9.2.3-1.

Table 6.3.1.9.2.3-1: Subscribe operation output parameters

Attribute name	Cardinality	Data type	Description
subscriptionId	1	String	Identifier of the subscription realized.

6.3.1.9.2.4 Operation results

After a successful subscription, the subscriber (such as OSS or MEO) will be registered to receive notifications of application lifecycle changes, as well as creation/deletion of application instance identifiers, with their associated application instances.

The result of the subscribe request shall indicate if the subscription has been successful or not with a standard success/error result. For a particular subscription, only notifications matching the filter will be delivered to the subscriber.

6.3.1.9.3 Notify

6.3.1.9.3.1 Description

The notify operation notifies a subscriber about events related to application lifecycle management operational state changes, as well as events related to the creation/deletion of application instance identifiers, with their associated application instances.

This operation distributes notifications to subscribers, and is a one-way operation issued only by the producer. In order to receive notifications, a consumer (such as OSS or MEO) has to perform an explicit subscribe operation beforehand.

Table 6.3.1.9.3.1-1 lists the information flow exchanged between OSS and MEO, MEPM and MEO.

Table 6.3.1.9.3.1-1: Notify operation

Message	Requirement	Direction
Notify	Mandatory	MEPM → MEO, or MEO → OSS

The following notifications may be sent by this operation, which are not further specified in the present document:

- AppLifecycleChangeNotification.
- AppInstanceIdentifierCreationNotification.
- AppInstanceIdentifierDeletionNotification.

6.3.1.9.4 Query subscription

6.3.1.9.4.1 Definition

The query operation enables a subscriber (such as OSS or MEO) to query subscription(s) for application instance operational state change notification. Table 6.3.1.9.4.1-1 lists the information flow exchanged between OSS and MEO, or MEO and MEPM.

Table 6.3.1.9.4.1-1: Query subscription operation

Message	Requirement	Direction
QuerySubscriptionRequest	Mandatory	OSS → MEO; or MEO → MEPM
QuerySubscriptionResponse	Mandatory	MEO → OSS; or MEPM → MEO

6.3.1.9.4.2 Input parameters

The input parameters shall follow the information in table 6.3.1.9.4.2-1.

Table 6.3.1.9.4.2-1: Query subscription operation input parameters

Attribute name	Cardinality	Data type	Description
n/a			

6.3.1.9.4.3 Output parameters

The output parameters returned by the operation shall follow the information in table 6.3.1.9.4.3-1.

Table 6.3.1.9.4.3-1: Query subscription operation output parameters

Attribute name	Cardinality	Data type	Description
appInstSubscriptionLinkList	1	SubscriptionLinkList	A list of subscriptions to application instances.

6.3.1.9.4.4 Operation results

The result of this operation shall indicate whether the query request is success or not. If successful, the information of subscription(s) shall be delivered to the requester (OSS or MEO).

6.3.1.9.5 Delete subscription operation

6.3.1.9.5.1 Definition

The delete operation deletes subscription(s) to application lifecycle management notification. Table 6.3.1.9.5.1-1 lists the information flow exchanged between OSS and MEO, or MEO and MEPM.

Table 6.3.1.9.5.1-1: Delete subscription operation

Message	Requirement	Direction
DeleteSubscriptionRequest	Mandatory	OSS → MEO; or MEO → MEPM
DeleteSubscriptionResponse	Mandatory	MEO → OSS; or MEPM → MEO

6.3.1.9.5.2 Input parameters

The input parameters of this operation shall follow the information in table 6.3.1.9.5.2-1.

Table 6.3.1.9.5.2-1: Delete subscription operation input parameters

Attribute name	Cardinality	Data type	Description
subscriptionId	1..N	String	The identifier(s) of application instance subscription(s).

6.3.1.9.5.3 Output parameters

The output parameters returned by the operation shall follow the information in table 6.3.1.9.5.3-1.

Table 6.3.1.9.5.3-1: Delete subscription operation output parameters

Attribute name	Cardinality	Data type	Description
n/a			

6.3.1.9.5.4 Operation results

The result of this operation shall indicate whether the delete request is success or not.

6.3.2 Void

6.3.3 Application package management interface

6.3.3.1 Fetch onboarded application package operation

6.3.3.1.1 Definition

This operation enables the MEPM to fetch onboarded application package. Table 6.3.3.1.1-1 lists the information flow exchanged between the MEO and the MEPM.

Table 6.3.3.1.1-1: Fetch onboarded package operation

Message	Requirement	Direction
FetchAppPackageRequest	Mandatory	OSS → MEO or MEPM → MEO
FetchAppPackageResponse	Mandatory	MEO → OSS or MEO → MEPM

6.3.3.1.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.1.2-1.

Table 6.3.3.1.2-1: Fetch onboarded package operation input parameters

Attribute name	Cardinality	Data type	Description
appPkgId	1	String	Identifier of the onboarded application package to be fetched.

6.3.3.1.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.3.1.3-1.

Table 6.3.3.1.3-1: Fetch onboarded package operation output parameters

Attribute name	Cardinality	Data type	Description
appPackage	0..1	Binary	The application package.

6.3.3.1.4 Operation results

After success operation, the MEO has provided to the requester a copy of the requested application package.

6.3.3.2 Query application package information operation

6.3.3.2.1 Definition

This interface allows the MEPM to query information about the Application Package. Table 6.3.3.2.1-1 lists the information flow exchanged between the MEO and the MEPM, and between the OSS and the MEO.

Table 6.3.3.2.1-1: Query application package operation

Message	Requirement	Direction
QueryAppPkgInfoRequest	Mandatory	OSS → MEO or MEPM → MEO
QueryAppPkgInfoResponse	Mandatory	MEO → OSS or MEO → MEPM

6.3.3.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.2.2-1.

Table 6.3.3.2.2-1: Query application package operation input parameters

Attribute name	Cardinality	Data type	Description
filter	1	Filter	Filter defining the application packages on which the query applies, based on attributes of the application package. It may also be used to specify one or more application packages to be queried by providing their identifiers. See note.
attributeSelector	0..N	String	It provides a list of attribute names of the application package. If present, only these attributes will be returned for the application package matching the filter. If absent, the complete application package will be returned.
NOTE: See table 7.3.1.3.2-1 for the attribute-based filter and selector.			

6.3.3.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.3.2.3-1.

Table 6.3.3.2.3-1: Query application package operation output parameters

Attribute name	Cardinality	Data type	Description
queryResult	0..N	AppPkgInfo	Details of the application packages matching the attribute filter. If an attribute selector is present, only the attributes listed in attribute elector will be returned for the selected entities.

6.3.3.2.4 Operation results

After successful operation, the MEO has queried the internal application package information objects. The result of the operation indicates if it has been successful or not with a standard success/error result. For a particular query, information about the application package that the consumer has access to and that are matching the filter shall be returned.

6.3.3.3 Subscribe operation

6.3.3.3.1 Definition

This operation enables the OSS or MEPM to subscribe with a filter for the notifications related to events of application packages sent by the MEO. Table 6.3.3.3.1-1 lists the information flow exchanged between the OSS and MEO, or the MEO and the MEPM.

Table 6.3.3.3.1-1: Subscribe operation

Message	Requirement	Direction
SubscribeRequest	Mandatory	OSS → MEO or MEPM → MEO
SubscribeResponse	Mandatory	MEO → OSS or MEO → MEPM

6.3.3.3.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.3.2-1.

Table 6.3.3.3.2-1: Subscribe operation input parameters

Attribute name	Cardinality	Data type	Description
inputFilter	1	Filter	Input filter for selecting the application package(s) and the related events notifications to subscribe to. This filter may contain information about specific types of events to subscribe to, or attributes of the application package.

6.3.3.3.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.3.3.3-1.

Table 6.3.3.3.3-1: Subscribe operation output parameters

Attribute name	Cardinality	Data type	Description
subscriptionId	1	String	Identifier of the subscription realized.

6.3.3.3.4 Operation results

After successful subscription, the OSS or MEPM is registered to receive notifications related to events of application packages sent by the MEO. The result of the operation shall indicate if the subscription has been successful or not with a standard success/error result. For a particular subscription, only notifications matching the filter will be delivered to the OSS or MEPM.

6.3.3.4 Notify application package operation

6.3.3.4.1 Definition

This operation distributes notifications to subscribers and can only be invoked as an operation by the MEO.

In order to receive notifications, the OSS or MEPM shall have a subscription.

The following notifications shall be notified/sent to subscribers by this operation:

- AppPackageOnBoardingNotification.
- AppPackageStateChangeNotification.

The format of both notification is the AppPkgNotification type specified in clause 6.2.3.6.2.

Table 6.3.3.4.1-1: Notify operation

Message	Requirement	Direction
Notify	Mandatory	MEO →OSS or MEO → MEPM

6.3.3.5 Onboarding operation

6.3.3.5.1 Definition

This operation will onboard an application package in the MEO.

Table 6.3.3.5.1-1 lists the information flow exchanged between the OSS and the MEO.

Table 6.3.3.5.1-1: Onboarding operation

Message	Requirement	Direction
OnboardAppPkgRequest	Mandatory	OSS → MEO
OnboardAppPkgResponse	Mandatory	MEO → OSS

6.3.3.5.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.5.2-1.

Table 6.3.3.5.2-1: Onboarding operation input parameters

Attribute name	Cardinality	Data type	Description
name	1	String	Name of the application package to be onboarded.
version	1	String	Version of the application package to be onboarded.
provider	1	String	Provider of the application package to be onboarded.
checksum	1	Not specified.	Checksum of the on-boarded application package.
userDefinedData	0..N	KeyValuePair	User defined data for the application package. See note 1.
appPackagePath	1	URI	Address information based on which the application package may be obtained. See note 2.
NOTE 1: This data type is not specified in the present document. NOTE 2: This Structure may be the address information related to an FTP server when the application package is stored, or be a URI where the MEO may download the application package.			

6.3.3.5.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.3.5.3-1.

Table 6.3.3.5.3-1: Onboarding operation output parameters

Attribute name	Cardinality	Data type	Description
appPkgId	1	String	Identifier of the on-boarded the Application package.
appDId	1	String	Identifier that identifies the application package in a globally unique way. See note.
NOTE: This identifier, which is managed by the application provider, identifies the application package and the AppD in a globally unique way.			

6.3.3.5.4 Operation results

The result of the operation indicates whether the on-boarding of the application package has been successful or not with a standard success/error result.

The appPkgId of onboarded application package will only be returned when the operations has been successful.

Once on-boarded, the application package will be known to and validated by the MEO. It will be in "Enabled, Not in use" state, allowing its use for application lifecycle management. For details of state model of application, refer to clause A.2.

6.3.3.6 Enable operation

6.3.3.6.1 Definition

This operation will enable a previously disabled application package, allowing again its use for instantiation of new application instances. The "In use/Not in use" sub-state shall not change as a result of the operation.

Table 6.3.3.6.1-1 lists the information flow exchanged between the OSS and the MEO.

Table 6.3.3.6.1-1: Enable operation

Message	Requirement	Direction
EnableAppPkgRequest	Mandatory	OSS → MEO
EnableAppPkgResponse	Mandatory	MEO → OSS

6.3.3.6.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.6.2-1.

Table 6.3.3.6.2-1: Enable operation input parameters

Attribute name	Cardinality	Data type	Description
appPkgId	1	String	Identifier of the on-boarded application package.

6.3.3.6.3 Output parameters

No output parameter.

6.3.3.6.4 Operation results

The result of the operation indicates if it has been successful or not with a standard success/error result.

If the application was already enabled, this operation will return an error.

6.3.3.7 Disable operation

6.3.3.7.1 Definition

This operation will disable a previously enabled application package, preventing any further use for instantiation of new network application instance with this application package. The "In use/Not in use" sub-state shall not change as a result of the operation. After an application package is disabled successfully, the state of this application package is "disabled, not in use" or "disabled, in use", see clause A.2 for the state of the application package.

Table 6.3.3.7.1-1 lists the information flow exchanged between the OSS and the MEO.

Table 6.3.3.7.1-1: Disable operation

Message	Requirement	Direction
DisableAppPkgRequest	Mandatory	OSS → MEO
DisableAppPkgResponse	Mandatory	MEO → OSS

6.3.3.7.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.7.2-1.

Table 6.3.3.7.2-1: QueryAppPackage operation input parameters

Attribute name	Cardinality	Data type	Description
onboardedAppPkgId	1	String	Identifier of the on-boarded application package.

6.3.3.7.3 Output parameters

No output parameter.

6.3.3.7.4 Operation results

The result of the operation indicates if it has been successful or not with a standard success/error result.

If the application package was already disabled, this operation will return an error.

6.3.3.8 Void

6.3.3.9 Delete operation

6.3.3.9.1 Definition

This operation will delete one application package.

An application package shall only be deleted when it is disabled and there is no instantiated application instance using it.

Table 6.3.3.9.1-1 lists the information flow exchanged between the OSS and the MEO.

Table 6.3.3.9.1-1: Delete operation

Message	Requirement	Direction
DeleteAppPkgRequest	Mandatory	OSS → MEO
DeleteAppPkgResponse	Mandatory	MEO → OSS

6.3.3.9.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.9.2-1.

Table 6.3.3.9.2-1: Delete application package operation input parameters

Attribute name	Cardinality	Data type	Description
appPkgId	1	String	Identifier of information held by the MEO about the specific on-boarded application package, which is to be deleted. This identifier was allocated by the MEO.

6.3.3.9.3 Output parameters

No output parameter.

6.3.3.9.4 Operation results

The result of the operation indicates if it has been successful or not with a standard success/error result.

6.3.3.10 Abort application package deletion operation

This operation is no longer supported.

6.3.3.11 Query subscription operation

6.3.3.11.1 Definition

This operation enables the OSS or MEPM to query subscription(s) for events of application packages sent by the OSS or MEO. Table 6.3.3.11.1-1 lists the information flow exchanged between the OSS and the MEO, or the MEO and the MEPM.

Table 6.3.3.11.1-1: Query subscription operation

Message	Requirement	Direction
QuerySubscriptionRequest	Mandatory	OSS → MEO or MEPM → MEO
QuerySubscriptionResponse	Mandatory	MEO → OSS or MEO → MEPM

6.3.3.11.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.11.2-1.

Table 6.3.3.11.2-1: Query subscription operation input parameters

Attribute name	Cardinality	Data type	Description
n/a			

6.3.3.11.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.3.11.3-1.

Table 6.3.3.11.3-1: Query subscription operation output parameters

Attribute name	Cardinality	Data type	Description
appPkgSubscriptionInfo	0 .. N	AppPkgSubscriptionInfo	A list of application package subscriptions

6.3.3.11.4 Operation results

The result of this operation shall indicate whether the query request is success or not. If successful query, the information of subscription(s) shall be delivered to the requester OSS or MEPM.

6.3.4 Granting interface

6.3.4.1 Introduction

This interface allows the MEPM to obtain from the MEO permission and configuration parameters for an application lifecycle operation. Further, this interface allows to retrieve the granting result.

6.3.4.2 Granting request

6.3.4.2.1 Definition

The MEPM sends the granting request for permission on an operation of application instance. Table 6.3.4.2.1-1 lists the information flow for the granting request.

Table 6.3.4.2.1-1: Granting request

Message	Requirement	Direction
Granting request	Mandatory	MEPM → MEO
Granting response	Mandatory	MEO → MEPM

6.3.4.2.2 Input parameters

The input parameters for the granting request is shown in table 6.3.4.2.2-1.

Table 6.3.4.2.2-1: Input parameters of granting request

Attribute name	Cardinality	Data type	Description
grantRequest	1	GrantRequest	The parameters for the granting request.

6.3.4.2.3 Output parameters

The output parameters shall follow the indications in table 6.3.4.2.3-1.

Table 6.3.4.2.3-1: Output parameters of granting request

Attribute name	Cardinality	Data type	Description
grant	1	Grant	Result of the granting request

6.3.4.2.4 Operation results

In case of success, the MEO returns the granting result in the grant message.

7 API definitions

7.1 Introduction

This clause defines the RESTFul resources and operations over reference point Mm1 and Mm3 APIs for:

- application package management; and
- application life cycle management.

7.2 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 "app_pkgm" for application package management interface, or "app_lcm" for application life cycle management interface.
- The "apiVersion" shall be set to "v1" for the present document. All resource URIs in the sub-clauses below is defined relative to the above root URI.

Due to the specific structure how application packages are identified, there are two resource sub-trees with identical structure provided which only differ in the identifier per individual application package resource. Application packages can be identified by a MEO-managed identifier known as appPkgId which is assigned during the application package onboarding process, or by an identifier known as appDId defined by the application vendor during application package creation. The set of packages identified by the appDId is a subset of the application packages identified by the appPkgId, containing all those packages that have completed their onboarding process and are available for use by the MEPM.

For any given appDId value, there shall be at most one associated appPkgId value in the whole resource tree visible to the MEPM.

Figure 7.2-1 illustrates the resource URI structure of application package management on the reference point of Mm1 or Mm3.

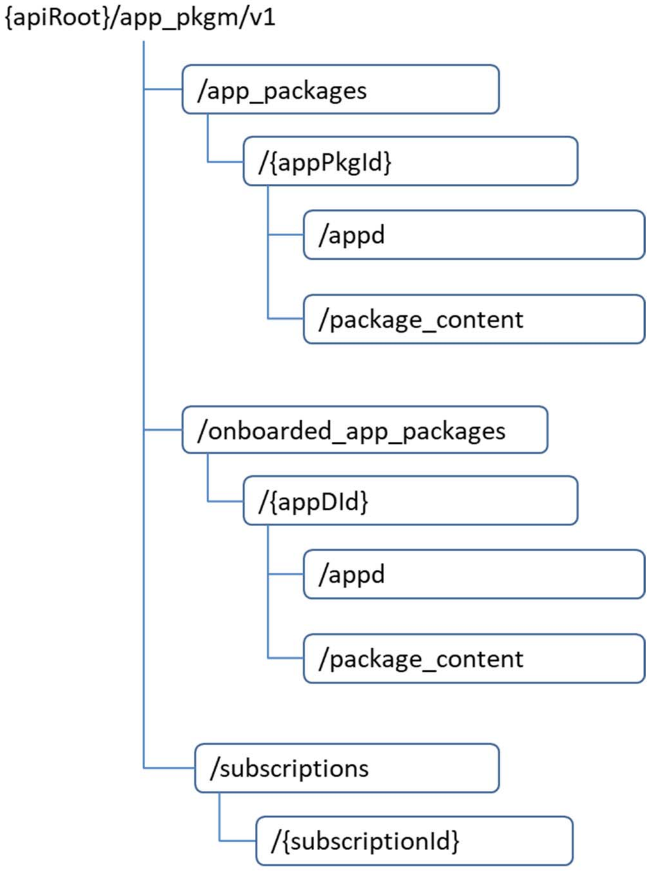

Figure 7.2-1: The resource URI structure of application package management

Figure 7.2-2 illustrates the resource URI structure of application life cycle management interface on the reference point of Mm1, or Mm3.

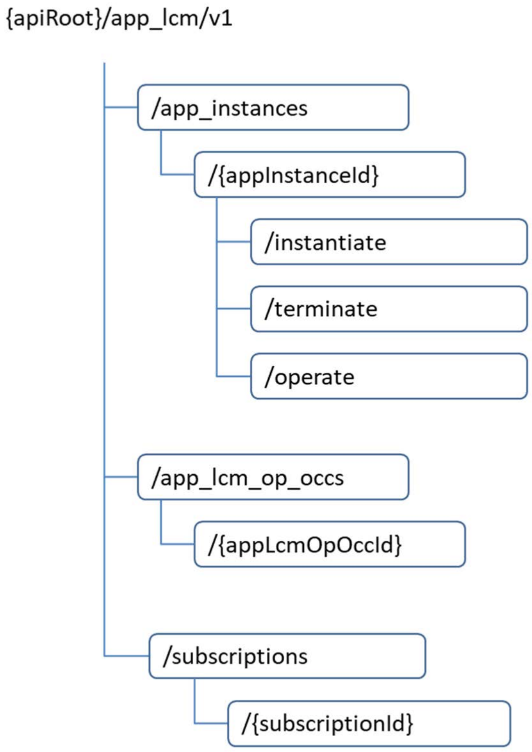

Figure 7.2-2: The resource URI structure of application life cycle management

Figure 7.2-3 illustrates the resource URI structure of granting on Mm3.

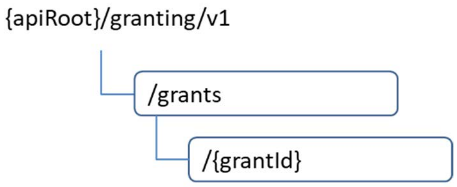

Figure 7.2-3: The resource URI structure of granting

Table 7.2-1 summarizes the resources and associated HTTP methods for application package management API over Mm1 reference points.

Table 7.2-1: Overview of resources and methods of MEO's application package management on Mm1

Resource name	Resource URI	HTTP Method	Description
Application packages	/app_packages /onboarded_app_packages	POST	Create a new resource for on-boarded application package.
		GET	Query on-boarded application package information.

Individual application package	/app_packages/{appPkgId}
	/onboarded_app_packages/{appDId}
		GET	Read information of an individual on-boarded application package.

		PATCH	Enable or disable an individual on-boarded application package.
		DELETE	Delete an individual on-boarded application package.
Application descriptor	/app_packages/{appPkgId}/appd /onboarded_app_packages/{appDId}/appd	GET	Read application descriptor of an onboarded application package.

Resource name	Resource URI	HTTP Method	Description
Application package content	/app_packages/{appPkgId}/package_content /onboarded_app_package/{appDId}/package_content	GET	Fetch an on-boarded application package.
		PUT	Upload an application package by providing the content of application package.
Subscriptions	/subscriptions	POST	Subscribe to notification related to on-boarding and/or changes of application packages.
		GET	Query multiple subscriptions
Individual subscription	/subscriptions/{subscriptionId}	GET	Read resource of an individual subscription.
		DELETE	Terminate an individual subscription.
Notification endpoint	(client provided)	POST	Notify application package on-boarding or change.

Table 7.2-2 summarizes the resources and associated HTTP methods for application life cycle management APIs over Mm1 reference points.

Table 7.2-2: Overview of resources and methods of MEO's application life cycle management on Mm1

Resource name	Resource URI	HTTP Method	Description
Application instances	/app_instances	POST	Create an application instance resource.
		GET	Query multiple application instance resources.
Individual application instance	/app_instances/{appInstanceId}	GET	Read an application instance resource.
		DELETE	Delete individual application instance resource.
Instantiate application instance task	/app_instances/{appInstanceId}/instantiate	POST	Instantiate the application instance.
Terminate application instance task	/app_instances/{appInstanceId}/terminate	POST	Terminate the application instance.
Operate application instance task	/app_instances/{appInstanceId}/operate	POST	Start or stop the application instance.
Application LCM operation occurrences	/app_lcm_op_occs	GET	Query multiple application lifecycle operation occurrences.
Individual application LCM operation occurrence	/app_lcm_op_occs/{appLcmOpOccId}	GET	Read the operation state of the individual application lifecycle operation occurrence.
Subscriptions	/subscriptions	POST	Subscribe to notifications related to application instances' lifecycle change.
		GET	Query multiple subscriptions.
Individual subscription	/subscriptions/{subscriptionId}	GET	Read an individual subscription resource.
		DELETE	Terminate an individual subscription.
Notification endpoint	(client provided)	POST	Notify about application instance's lifecycle change.

Table 7.2-3 summarizes the resources and associated HTTP methods for MEO's application package management APIs over Mm3 reference points.

Table 7.2-3: Overview of resources and methods of MEO's application package management on Mm3

Resource name	Resource URI	HTTP Method	Description
Application packages	/app_packages /onboarded_app_packages	GET	Query information about multiple on-boarded application packages.

Individual application package	/app_packages/{appPkgId}
	/onboarded_app_packages/{appDId}
		GET	Read information about individual on-boarded application package.

Application package content	/app_packages/{appPkgId}/package_content /onboarded_app_package/{appDId}/package_content	GET	Fetch an on-boarded application package.
Application descriptor	/app_packages/{appPkgId}/appd /onboarded_app_packages/{appDId}/appd	GET	Read the application descriptor of the on-boarded application package.
Subscriptions	/subscriptions	POST	Subscribe to notification related to on-boarding and/or changes of application packages.
		GET	Query multiple subscriptions.
Individual subscription	/subscriptions/{subscriptionId}	GET	Read an individual subscription.
		DELETE	Terminate an individual subscription.
Notification endpoint	(client provided)	POST	Notify application package on-boarding or change.

Table 7.2-4 summarizes the resources and associated HTTP methods for MEPM's application life cycle management API over Mm3 reference points.

Table 7.2-4: Overview of resources and methods of MEPM's application life cycle management on Mm3

Resource name	Resource URI	HTTP Method	Description
Application instances	/app_instances	POST	Create an application instance resource
		GET	Query multiple application instances
Individual application instance	/app_instances/{appInstanceId}	GET	Read application instance
		DELETE	Delete individual application instance
Instantiate application instance task	/app_instances/{appInstanceId}/instantiate	POST	Instantiate an application instance
Terminate application instance task	/app_instances/{appInstanceId}/terminate	POST	Terminate an application instance
Operate application instance task	/app_instances/{appInstanceId}/operate	POST	Start or stop an application instance
Application LCM operation occurrences	/app_lcm_op_occs	GET	Query multiple application lifecycle operation occurrences
Indiviual application LCM operation occurrence	/app_lcm_op_occs/{appLcmOpOccId}	GET	Read an individual application lifecycle management operation occurrence
Subscriptions	/subscriptions	POST	Subscribe to notifications related to application instance's lifecycle change
		GET	Query multiple subscriptions
Individual subscription	/subscriptions/{subscriptionId}	GET	Query an individual subscription
		DELETE	Terminate an individual subscription
Notification endpoint	(client provided)	POST	Notify about application instance's lifecycle change

Table 7.2-5 summarizes the resources and associated HTTP methods for MEO's application life cycle management API over Mm3 reference points.

Table 7.2-5: Overview of resources and methods of MEO's application life cycle management on Mm3

Resource name	Resource URI	HTTP Method	Description
Grants	/grants	POST	Request a grant for a particular app LCM operation
Individual grant	/grants/{grantId}	GET	Read the status of grant for the application LCM operation

7.3 Resources of application package management on Mm1 and Mm3

7.3.1 Resource: application packages

7.3.1.1 Description

This resource is used to represent application packages of data type "AppPkgInfo" specified in clause 6.2.

7.3.1.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_pkgm/v1/app_packages/

Resource URI variables for this resource are defined in table 7.3.1.2-1.

Table 7.3.1.2-1: Resource URI variables for the resource

Name	Definition
apiRoot	See clause 7.2

7.3.1.3 Resource methods

7.3.1.3.1 POST

The POST method is used to create a resource for on-boarding an application package to a MEO, which refers to the procedure of "onboarding operation" as described in clause 6.3.3.5.

The POST method is supported on Mm1 only.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.3.1.3.1-1 and 7.3.1.3.1-2.

Table 7.3.1.3.1-1: URI query parameters of POST method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.1.3.1-2: Data structures supported by POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	CreateAppPkg	1	The POST method is used to create a new resource for onboarding an application package onto a MEO.	
Response body	Data type	Cardinality	Response codes	Remarks
	AppPkgInfo	0..N	201 Created	Indicates a successful request. The response body shall contain a representation of the application package resource defined in clause 6.2. The HTTP response includes a "Location" HTTP header that contains the URI of the created resource.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.1.3.2 GET

This GET method queries information relating to on-boarded application packages in the MEO matching the filtering criteria. It refers to the procedure of "query operation" of application package as described in clause 6.3.3.2.

The GET method is supported on Mm1 and Mm3.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.3.1.3.2-1 and 7.3.1.3.2-2.

Table 7.3.1.3.2-1: URI query parameters of GET method on the resource

Attribute name	Cardinality	Description
filter	0..1	Attribute-based filtering parameters according to ETSI GS MEC 009 [4]. The API producer shall support receiving filtering parameters as part of the URI query string. All attribute names that appear in the AppPkgInfo and in data types referenced from it shall be supported in attribute-based filtering parameters. See clause 6.19 in ETSI GS MEC 009 [4] for details.
all_fields	0..1	Include all complex attributes in the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer shall support this parameter.
fields	0..1	Complex attributes of AppPkgInfo to be included into the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer should support this parameter.
exclude_fields	0..1	Complex attributes of AppPkgInfo to be excluded from the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes of AppPkgInfo from the response. The following attributes shall be excluded from the AppPkgInfo structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: • checksum; • softwareImanges; • additionalArtifacts.

Table 7.3.1.3.2-2: Data structures supported by GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response codes	Remarks
	AppPkgInfo	0..N	200 OK	Indicate the success of request. The response body shall contain a list of representations of the "individual application package" resources that match the attribute filter.
	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	1	400 Bad Request	Error: Invalid attribute-based filtering expression. In the returned ProblemDetails structure, the "detail" attribute shall convey more information about the error.
	ProblemDetails	1	400 Bad Request	Error: Invalid attribute selector. In the returned ProblemDetails structure, the "detail" attribute shall 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.

	Data type	Cardinality	Response codes	Remarks
Response body	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.1.3.3 PUT

Not supported.

7.3.1.3.4 DELETE

Not supported.

7.3.1.3.5 PATCH

Not supported.

7.3.2 Resource: individual application package

7.3.2.1 Description

This resource is used to represent an individual application package of data type "AppPkgInfo" specified in clause 6.2.

7.3.2.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_pkgm/v1/app_packages/{appPkgId}

Resource URI variables for this resource are defined in table 7.3.2.2-1.

Table 7.3.2.2-1: Resource URI variables for the resource

Name	Definition
apiRoot	See clause 7.2
appPkgId	Identifier of an individual application package resource

7.3.2.3 Resource methods

7.3.2.3.1 POST

Not supported.

7.3.2.3.2 GET

This GET method is used to query the information related to individual application package resources.

The GET method is supported on Mm1 and Mm3.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.3.2.3.2-1 and 7.3.2.3.2-2.

Table 7.3.2.3.2-1: URI query parameters of GET method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.2.3.2-2: Data structures supported by GET request/response on this resource

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

7.3.2.3.3 PUT

Not supported.

7.3.2.3.4 DELETE

This DELETE method realizes the procedure of "delete operation" of application package resource in MEO as described in clause 6.3.3.9.

The DELETE method is supported on Mm1 only.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.3.2.3.4-1 and 7.3.2.3.4-2, which refer to tables 6.3.3.9.2-1.

Table 7.3.2.3.4-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.2.3.4-2: 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 successful deletion of application package resource, an empty response body shall be returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.2.3.5 PATCH

This PATCH method updates the operational state of an individual application package resource used by the procedure of "enable operation" as described in clause 6.3.3.6, "disable operation" as described in clause 6.3.3.7.

The PATCH method is supported on Mm1 only.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.3.2.3.5-1 and 7.3.2.3.5-2.

Table 7.3.2.3.5-1: URI query parameters of PATCH method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.2.3.5-2: Data structures supported by PATCH request/response on this resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	AppPkgInfoModification	1	Parameters for application package information modification.	
Response body	**Data type**	**Cardinality**	**Response Codes**	**Remarks**
	AppPkgInfoModification	1	200 OK	Shall be returned when the operation has been completed 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.
	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	1	409 Conflict	Shall be returned upon the following errors: - The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to any of the following scenarios: • Disable an application package resource of which the operational state is not ENABLED. • Enable an application package resource of which the operational state is not DISABLED. - The operation of onboarding the application package has not completed. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.

	Data type	Cardinality	Response Codes	Remarks
Response body	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.3 Resource: subscriptions

7.3.3.1 Description

This resource is used to represent subscriptions to notifications about application package changes. The subscriber can use this resource to subscribe to notifications related to the application package management.

7.3.3.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_pkgm/v1/subscriptions.

Resource URI variables for this resource are defined in table 7.3.3.2-1.

Table 7.3.3.2-1: Resource URI variables for the resource

Name	Definition
apiRoot	See clause 7.2

7.3.3.3 Resource methods

7.3.3.3.1 POST

The POST method is used to subscribe to notifications about on-boarding an application package, or about operational state changes of on-boarded application package, which is mapped to the procedure of "subscription operation" as described in clause 6.3.3.3.

The POST method for subscriptions is supported on Mm1 and Mm3.

This method shall comply with the URI request and response data structures, and response codes, as specified in the table 7.3.3.3.1-1.

Table 7.3.3.3.1-1: Data structures supported by POST request/response on this resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>AppPkgSubscription</td><td>1</td><td colspan="2">The input parameters of "subscribe operation" to notifications about changes related to application package management for the on-boarding, or operational state change of application package.</td></tr>
<tr><th rowspan="8">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>AppPkgSubscriptionInfo</td><td>1</td><td>201 Created</td><td>Upon success, a response body representing the created subscription shall be returned.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>400 Bad Request</td><td>It is used to indicate that incorrect parameters were passed to the request.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>401 Unauthorized</td><td>It is used when the client did not submit credentials.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>1</td><td>403 Forbidden</td><td>The operation is not allowed given the current status of the resource.
More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>404 Not Found</td><td>It is used when a client provided a URI that cannot be mapped to a valid resource URI.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>406 Not Acceptable</td><td>It is used to indicate that the server cannot provide the any of the content formats supported by the client.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>429 Too Many Requests</td><td>It is used when a rate limiter has triggered.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
</table>

7.3.3.3.2 GET

This GET method is used to retrieve the information of subscriptions to individual application package resource in MEO. Upon success, the response contains the list of links to the subscriptions that are present for the requestor.

The GET method is supported on Mm1 and Mm3.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.3.3.3.2-1 and 7.3.3.3.2-2.

Table 7.3.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.3.3.2-2: 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**
	AppPkgSubscriptionLinkList	1	200 OK	Upon success, a response body containing a list of zero or more subscriptions shall be returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.3.3.3 PUT

Not supported.

7.3.3.3.4 DELETE

Not Supported.

7.3.3.3.5 PATCH

Not Supported.

7.3.4 Resource: individual subscription

7.3.4.1 Description

This resource is used to represent an individual subscription to notifications about application package changes, which is mapped to the procedure of "subscription operation" as described in clause 6.3.3.3.

7.3.4.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_pkgm/v1/subscriptions/{subscriptionId}.

Resource URI variables for this resource are defined in table 7.3.4.2-1.

Table 7.3.4.2-1: Resource URI variables for the resource

Name	Definition
apiRoot	See clause 7.2.
subscriptionId	Identifier of an individual subscription to notifications about application package changes.

7.3.4.3 Resource methods

7.3.4.3.1 POST

Not supported.

7.3.4.3.2 GET

This GET method is used to retrieve the individual subscription information to the application package resource in MEO.

The GET method is supported on Mm1 and Mm3.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.3.4.3.2-1 and 7.3.4.3.2-2.

Table 7.3.4.3.2-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.4.3.2-2: Data structures supported by GET request/response on this resource

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

7.3.4.3.3 PUT

Not supported.

7.3.4.3.4 DELETE

This DELETE method is used to delete the individual subscription to notifications about application package changes in MEO.

The DELETE method is supported on Mm1 and Mm3.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.3.4.3.4-1 and 7.3.4.3.4-2.

Table 7.3.4.3.4-1: URI query parameters supported by DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.4.3.4-2: Data structures supported by 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	The subscription resource was deleted successfully. The response body shall be empty.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.4.3.5 PATCH

Not supported.

7.3.5 Resource: notification endpoint

7.3.5.1 Description

This resource is used to represent a notification endpoint, which is mapped to the procedure of "notify application package operation" as described in clause 6.3.3.3. The API producer can use this resource to send notifications related to application package management events to a subscribed API consumer.

7.3.5.2 Resource definition

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

Resource URI variables for this resource are defined in table 7.3.5.2-1.

Table 7.3.5.2-1: Resource URI variables for the resource

Name	Definition
n/a	

7.3.5.3 Resource methods

7.3.5.3.1 POST

The POST method delivers a notification from the application package management resource in MEO to the subscriber.

The POST method is supported on Mm1 and Mm3.

This method shall follow the provisions specified in tables 7.3.5.3.1-1 and 7.3.5.3.1-2 for URI parameters, request and response data structures, and response codes.

Table 7.3.5.3.1-1: URI query parameters supported by POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.5.3.1-2: Data structures supported by POST request/response on this resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>AppPkgNotification</td><td>1</td><td colspan="2">A notification of an application package for on-boarding or operational state change.</td></tr>
<tr><th rowspan="6">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>n/a</td><td></td><td>204 No Content</td><td>The notification was delivered successfully.
The response body shall be empty.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>401 Unauthorized</td><td>It is used when the client did not submit credentials.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>1</td><td>403 Forbidden</td><td>The operation is not allowed given the current status of the resource.
More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>404 Not Found</td><td>It is used when a client provided a URI that cannot be mapped to a valid resource URI.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>429 Too Many Requests</td><td>It is used when a rate limiter has triggered.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
</table>

7.3.5.3.2 GET

Not supported.

7.3.5.3.3 PUT

Not supported.

7.3.5.3.4 DELETE

Not supported.

7.3.5.3.5 PATCH

Not supported.

7.3.6 Resource: application descriptor

7.3.6.1 Description

This resource represents an application descriptor "AppD" contained in an on-boarded application package. The client can use this resource to obtain the content of the AppD specified in clause 6.2 over either the Mm1 or Mm3 reference point.

7.3.6.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_pkgm/v1/app_packages/{appPkgId}/appd
- Resource URI: {apiRoot}/app_pkgm/v1/onboarded_app_packages/{appDId}/appd

Resource URI variables for this resource are defined in table 7.3.6.2-1.

Table 7.3.6.2-1: Resource URI variables for the resource

Name	Definition
apiRoot	See clause 7.2.
appPkgId	Identifier of an on-boarded individual application package.

7.3.6.3 Resource methods

7.3.6.3.1 POST

This method is not supported. When this method is requested on this resource, the "405 Method Not Allowed" response is returned.

7.3.6.3.2 GET

This GET method reads the content of the AppD of on-boarded individual application package resources. The format of the AppD is left for future specification.

The selection of the format is controlled by the "Accept" HTTP header passed in the GET request:

- If the "Accept" header contains only "text/plain" (with appropriate indication of character set) and the AppD is implemented as a single file, the file shall be returned; otherwise, an error message shall be returned.
- If the "Accept" header contains only "application/zip", the single file or the multiple files that make up the AppD shall be returned embedded in a ZIP file.
- If the "Accept" header contains both "text/plain" (with appropriate indication of character set) and "application/zip", it is up to the MEO to choose the format to return for a single-file AppD; for a multi-file AppD, a ZIP file shall be returned.

The default format of the ZIP file can be referred to ETSI GS NFV-SOL 004 [i.6] where only the YAML files representing the AppD, and information needed to navigate the ZIP file and to identify the file that is the entry point for parsing the AppD (such as TOSCA-meta or manifest files or naming conventions) are included.

The GET method is supported on Mm1 and Mm3.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.3.6.3.2-1 and 7.3.6.3.2-2.

Table 7.3.6.3.2-1: URI query parameters of GET method on the resource

Attribute name	Cardinality	Description
filter	0..1	Attribute-based filtering parameters according to ETSI GS MEC 009 [4]. The API producer shall support receiving filtering parameters as part of the URI query string. All attribute names that appear in the AppD and in data types referenced from it shall be supported in attribute-based filtering parameters. See clause 6.19 in ETSI GS MEC 009 [4] for detail.
all_fields	0..1	Include all complex attributes in the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer shall support this parameter.
fields	0..1	Complex attributes of AppD to be included into the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer should support this parameter.
exclude_fields	0..1	Complex attributes of AppD to be excluded from the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes of AppD from the response. The following attributes shall be excluded from the AppPkgInfo structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: - virtualComputeDescriptor; - swImageDescriptor; - virtualStorageDescriptor; - appExtCpd; - terminateAppInstanceOpConfig; - changeAppInstanceStateOpConfig.

Table 7.3.6.3.2-2: Data structures supported by GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response codes	Remarks
	AppD	1	200 OK	Indicates the success of request, and the content of the AppD is returned. The payload body shall contain a copy of the file representing the AppD or a ZIP file that contains the file or multiple files representing the AppD. The "Content-Type" HTTP header shall be set according to the format of the returned file, which is selected according to "Accept" HTTP header options passed in the request.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that the request contained either: • incorrect parameters; or • an invalid attribute-based filtering expression; or • an invalid attribute selector. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the specific 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	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.6.3.3 PUT

This method is not supported. When this method is requested on this resource, the "403 Forbidden" response is returned.

7.3.6.3.4 DELETE

This method is not supported. When this method is requested on this resource, the "405 Method Not Allowed" response is returned.

7.3.6.3.5 PATCH

Not supported.

7.3.7 Resource: application package content

7.3.7.1 Description

This resource represents an individual application package identified by the application package identifier, or application descriptor identifier allocated by the MEO. The client can use this resource to upload or fetch the content of the application package.

7.3.7.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_pkgm/v1/app_packages/{appPkgId}/package_content
- Resource URI: {apiRoot}/app_pgkm/v1/onboarded_app_packages/{appDId}/package_content

Resource URI variables for this resource are defined in table 7.3.7.2-1.

Table 7.3.7.2-1: Resource URI variables for the resource

Name	Definition
apiRoot	See clause 7.2
appPkgId	Identifier of an individual application package
appDId	Identifier of an application descriptor

7.3.7.3 Resource methods

7.3.7.3.1 POST

Not supported.

7.3.7.3.2 GET

The GET method is used to fetch the onboarded application package content identified by appPkgId or appDId. The client can use this resource to fetch the content of the application package.

The GET method is supported on Mm1 and Mm3.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.3.7.3.2-1 and 7.3.7.3.2-2.

Table 7.3.7.3.2-1: URI query parameters of GET method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.7.3.2-2: Data structures supported by GET request/response on this resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>n/a</td><td></td><td colspan="2">The request may contain a "Range" HTTP header to obtain single range of bytes from the application package file. This can be used to continue an aborted transmission.
If the MEO does not support range requests, it should return the whole file with a 200 OK response instead.</td></tr>
<tr><th rowspan="4">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>n/a</td><td>1</td><td>200 OK</td><td>On success, a copy of the on-boarded application package is returned in the response body.
The "Content-Type" HTTP header shall be set according to the type of the file, i.e. to "application/zip" for an application package.</td></tr>
<tr><td>n/a</td><td>1</td><td>206 Partial Content</td><td>On success, if the MEO supports range requests, a single consecutive byte range from the content of the application package file shall be returned.
The response body shall contain the requested part of the application package file.
The "Content-Range" HTTP header shall be provided according to IETF RFC 7233 [6].
The "Content-Type" HTTP header shall be set as defined above for the "200 OK" response.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>400 Bad Request</td><td>It is used to indicate that incorrect parameters were passed to the request.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
</table>

	Data type	Cardinality	Response codes	Remarks
Response body	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	416 Range Not Satisfiable	Shall be returned upon the following error: The byte range passed in the "Range" header did not match any available byte range in the application package file (e.g. "access after end of file"). The response body may contain a ProblemDetails structure.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.7.3.3 PUT

This PUT method uploads the content of application package.

The POST method is supported on Mm1 only.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.3.7.3.3-1 and 7.3.7.3.3-2.

Table 7.3.7.3.3-1: URI query parameters of PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.3.7.3.3-2: Data structures supported by PUT request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a		The payload body shall contain a ZIP file that represents the application package. The "Content-Type" HTTP header shall be set to "application/zip".	
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a		202 Accepted	The application package has been accepted for uploading, but the processing has not been completed. It is expected to take some time for processing. The response body shall be empty.
	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	1	409 Conflict	The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that the onboarding state of the application package resource is not CREATED. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.3.7.3.4 DELETE

Not supported.

7.3.7.3.5 PATCH

Not supported.

7.4 Resources of application lifecycle management on Mm1

7.4.1 Resource: application instances

7.4.1.1 Description

This resource represents application instances of data type "AppInstanceInfo" specified in clause 6.2. The consumer can use this resource to request resource allocation for application instance and query the information of resource.

7.4.1.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/app_instances.

Resource URI variables for this resource are defined in table 7.4.1.2-1.

Table 7.4.1.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2

7.4.1.3 Resource methods

7.4.1.3.1 POST

The POST method is used to create an application instance resource, which refers to the procedure of "creating application instance resource operation" described in clause 6.3.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.1.3.1-1 and 7.4.1.3.1-2.

Table 7.4.1.3.1-1: URI query parameters of POST method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.1.3.1-2: Data structures supported by POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	CreateAppInstance Request	1	The POST method is used to create an application instance resource.	
Response body	Data type	Cardinality	Response codes	Remarks
	AppInstanceInfo	1	201 Created	An application instance identifier and the related resource has been created successfully. The response body shall contain a representation of the created resource. 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.

	Data type	Cardinality	Response codes	Remarks
Response body	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.1.3.2 GET

The GET method retrieves information about the application instances resources. This refers to the procedure of "query application instance information operation" as described in clause 6.3.1.5.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.1.3.2-1 and 7.4.1.3.2-2.

Table 7.4.1.3.2-1: URI query parameters of GET method on the resource

Name	Cardinality	Description
filter	0..1	Attribute-based filtering parameters according to ETSI GS MEC 009 [4]. The API producer shall support receiving filtering parameters as part of the URI query string. All attribute names that appear in the AppInstanceInfo and in data types referenced from it shall be supported in attribute-based filtering parameters. See clause 6.19 in ETSI GS MEC 009 [4] for details.
all_fields	0..1	Include all complex attributes in the response. See clause 6.18 in ETSI GS MEC 009 [4] 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 009 [4] 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 [4] 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 009 [4] for details. The API producer shall support this parameter. The following attributes shall be excluded from the AppInstanceInfo structure in the response body if this parameter is provided, or none of the parameters "all_fields," "fields", "exclude_fields", "exclude_default" are provided: - vimConnectionInfo; - instantiate; - terminate; - operate.

Table 7.4.1.3.2-2: Data structures supported by 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**
	AppInstanceInfo	0..N	200 OK	Information about zero or more application instances was queried successfully. The response body shall contain in an array the representations of zero or more application instances.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.1.3.3 PUT

Not supported.

7.4.1.3.4 DELETE

Not supported.

7.4.1.3.5 PATCH

Not supported.

7.4.2 Resource: individual application instance

7.4.2.1 Description

This method is to retrieve information about an individual application instance. The client can use this resource to read the information of, or delete the individual application instance.

7.4.2.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/app_instances/{appInstanceId}.

Resource URI Variables for this resource are defined in table 7.4.2.2-1.

Table 7.4.2.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2
appInstanceId	Identifier of an individual application instance

7.4.2.3 Resource methods

7.4.2.3.1 POST

Not supported.

7.4.2.3.2 GET

The GET method retrieves the information of an individual application instance via reading an individual application instance resource, which is used by the procedure of "query application instance information operation" as described in clause 6.3.1.5.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.2.3.2-1 and 7.4.2.3.2-2.

Table 7.4.2.3.2-1: URI query parameters of GET method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.2.3.2-2: Data structures supported by GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a	0		
Response body	Data type	Cardinality	Response codes	Remarks
	AppInstanceInfo	1	200 OK	Information about an individual application instance was read successfully. The response body shall contain a representation of the read resource.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

Response body	Data type	Cardinality	Response codes	Remarks
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.2.3.3 PUT

Not supported.

7.4.2.3.4 DELETE

The DELETE method deletes an individual application instance resource, which refers to the procedure of "delete application instance identifier operation" as described in clause 6.3.1.8.

The method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.2.3.4-1 and 7.4.2.3.4-2.

Table 7.4.2.3.4-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.2.3.4-2: 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	The application instance resource and the associated application instance identifier were deleted successfully. The response body shall be empty.
	ProblemDetails	1	409 Conflict	The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is due to the fact that the application instance resource is in INSTANTIATED state. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.2.3.5 PATCH

Not supported.

7.4.3 Resource: subscriptions

7.4.3.1 Description

This resource represents subscriptions to notifications of related to an application instance . The subscriber can use this resource to subscribe to notifications related to the application instance related changes, such as application instance operational state change or application LCM operation occurrence state change.

When this resource represents a subscription to notifications regarding application instance operational state changes, it shall follow the data type of "AppInstSubscriptionRequest" as specified in clause 6.2.2.12. The notifications related to this subscription shall follow the data type of "AppInstNotification" as specified in clause 6.2.2.11.

When this resource represents a subscription to the notifications regarding to application instance LCM operation occurrence state change, it shall follow the data type of "AppLcmOpOccSubscriptionRequest" as specified in clause 6.2.2.14. The notifications related to this subscription shall follow the data type of "AppLcmOpOccNotification" as specified in clause 6.2.2.16.

7.4.3.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/subscriptions.

Resource URI Variables for this resource are defined in table 7.4.3.2-1.

Table 7.4.3.2-1: Resource URI variables for the resource

Name	Definition
apiRoot	See clause 7.2

7.4.3.3 Resource methods

7.4.3.3.1 POST

The POST method is to subscribe to the notification of application instance operational state change, which is mapped to the procedure of "subscription operation".

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.4.3.3.1-1 and 7.4.3.3.1-2.

Table 7.4.3.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
subscriptionType	String	1	Query parameter to filter on a specific subscription type. Permitted values: • "AppInstanceStateChange". • "AppLcmOpOccStateChange".

Table 7.4.3.3.1-2: Data structures supported by the POST request/response on this resource

	Data type	Cardinality	Remarks	
Request body	{SubscriptionRequest}	1	The entity body in the subscription request contains the data type of the specific subscription to application instance state change event that is to be created: • AppInstSubscriptionRequest. • AppLcmOpOccSubscriptionRequest.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	{SubscriptionInfo}	1	201 Created	Upon success, a response body representing the created subscription is returned. The response body shall contain a representation of the created SubscriptionInfo with the appropriate data type: • AppInstSubscriptionInfo. • AppLcmOpOccSubscriptionInfo. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created subscription resource.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.3.3.2 GET

The GET method retrieves the information of multiple subscriptions to notifications related to an application instance.

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

Table 7.4.3.3.2-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a subscriptionType	String	0..1	Query parameter to filter on a specific subscription type. Permitted values: • "AppInstanceStateChange". • AppLcmOpOccStateChange".

Table 7.4.3.3.2-2: 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 a list of all subscriptions is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.3.3.3 PUT

Not supported.

7.4.3.3.4 DELETE

Not Supported.

7.4.3.3.5 PATCH

Not supported.

7.4.4 Resource: individual subscription

7.4.4.1 Description

This resource represents an individual subscription to notifications related to an application instance.

7.4.4.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/subscriptions/{subscriptionId}.

Resource URI variables for this resource are defined in table 7.4.4.2-1.

Table 7.4.4.2-1: Resource URI variables for the resource

Name	Definition
apiRoot	See clause 7.2.
subscriptionId	Represents an individual subscription to notification related to an application instance.

7.4.4.3 Resource methods

7.4.4.3.1 POST

Not supported.

7.4.4.3.2 GET

The GET method retrieves the individual subscription information by reading an individual subscription resource. This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.4.4.3.2-1 and 7.4.4.3.2-2.

Table 7.4.4.3.2-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
subscriptionType	String	1	Query parameter to filter on a specific subscription type. Permitted values: • "AppInstanceStateChange". • "AppLcmOpOccStateChange".

Table 7.4.4.3.2-2: 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**
	{SubscriptionInfo}	1	200 OK	Upon success, a response body containing a representation of the individual subscription resource shall be returned. The allowed data types for the SubscriptionInfo are: • AppInstSubscriptionInfo. • AppLcmOpOccSubscriptionInfo.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the ProblemDetails structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.4.3.3 PUT

Not supported.

7.4.4.3.4 DELETE

The DELETE method is used to delete an individual subscription to notifications related to application instances.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.4.4.3.4-1 and 7.4.4.3.4-2.

Table 7.4.4.3.4-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.4.3.4-2: 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	The subscription resource was deleted successfully. The response body shall be empty.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the ProblemDetails structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.4.3.5 PATCH

Not supported.

7.4.5 Resource: notification endpoint

7.4.5.1 Description

This resource represents a notification endpoint, which is mapped to the procedure of "notify application instance operational state change", or "notify application LCM operation occurrence state change". The producer can use this resource to send notifications related to application instance management events to a subscribed consumer.

7.4.5.2 Resource definition

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

Resource URI variables for this resource are defined in table 7.4.5.2-1.

Table 7.4.5.2-1: Resource URI variables for the resource

Name	Definition
n/a	

7.4.5.3 Resource methods

7.4.5.3.1 POST

The POST method delivers a notification from the application lifecycle management resource to the subscriber.

This method shall follow the provisions specified in tables 7.4.5.3.1-1 and 7.4.5.3.1-2 for URI parameters, request and response data structures, and response codes.

Table 7.4.5.3.1-1: URI query parameters supported by POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.5.3.1-2: Data structures supported by POST request/response on this resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>{Notification}</td><td>1</td><td colspan="2">A notification of an event related to an application instance.
The allowed data types of application state change notification are:
• AppInstNotification.
• AppLcmOpOccNotification.</td></tr>
<tr><th rowspan="6">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>n/a</td><td></td><td>204 No Content</td><td>The notification was delivered successfully. The response body shall be empty.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>401 Unauthorized</td><td>It is used when the client did not submit credentials.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>1</td><td>403 Forbidden</td><td>The operation is not allowed given the current status of the resource.
More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>404 Not Found</td><td>It is used when a client provided a URI that cannot be mapped to a valid resource URI.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>429 Too Many Requests</td><td>It is used when a rate limiter has triggered.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
</table>

7.4.5.3.2 GET

Not supported.

7.4.5.3.3 PUT

Not supported.

7.4.5.3.4 DELETE

Not supported.

7.4.5.3.5 PATCH

Not supported.

7.4.6 Resource: instantiate application instance task

7.4.6.1 Description

This resource represents the task of instantiating an application instance. The client can use this resource to instantiate an application instance.

7.4.6.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/app_instances/{appInstanceId}/instantiate

Resource URI variables for this resource are defined in table 7.4.6.2-1.

Table 7.4.6.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2.
appInstanceId	The identifier of the application instance. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating the new application instance resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

7.4.6.3 Resource methods

7.4.6.3.1 POST

The POST method is to instantiate the application instance.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.6.3.1-1 and 7.4.6.3.1-2.

Table 7.4.6.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.6.3.1-2: Data structures supported by a POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	InstantiateAppRequest	1	The payload body in the request contains the information necessary to instantiate an application instance.	
Response body	Data type	Cardinality	Response codes	Remarks
	n/a		202 Accepted	The request was accepted for processing, but the processing has not yet been completed. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "application LCM operation occurrence" resource that corresponds to this application instance instantiation operation.
	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 the appropriate 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	409 Conflict	The operation cannot currently be executed due to a conflict with the state of the resource. Typically, this is because the application instance resource is not in NOT_INSTANTIATED state. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.6.3.2 GET

Not supported.

7.4.6.3.3 PUT

Not supported.

7.4.6.3.4 DELETE

Not supported.

7.4.6.3.5 PATCH

Not supported.

7.4.7 Resource: terminate application instance task

7.4.7.1 Description

This resource represents the task of terminating an application instance. The client can use this resource to terminate an application instance.

7.4.7.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/app_instances/{appInstanceId}/terminate

Resource URI variables for this resource are defined in table 7.4.7.2-1.

Table 7.4.7.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2.
appInstanceId	The identifier of the application instance. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating the new application instance resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

7.4.7.3 Resource methods

7.4.7.3.1 POST

The POST method is used to terminate an application instance.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.7.3.1-1 and 7.4.7.3.1-2.

Once the MEPM has successfully completed the underlying application instance LCM operation occurrence, it shall set the "instantiationState" attribute in the representation of the "individual application instance" resource to the value "NOT_INSTANTIATED".

Table 7.4.7.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.7.3.1-2: Data structures supported by a POST request/response on this resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>TerminateAppRequest</td><td>1</td><td colspan="2">Parameters for the termination, as defined in clause 6.2.2.9.</td></tr>
<tr><th rowspan="9">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>n/a</td><td></td><td>202 Accepted</td><td>The request was accepted for processing, but the processing has not yet been completed.
The response body shall be empty.
The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "application LCM operation occurrence" resource that corresponds to this operation.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>400 Bad Request</td><td>It is used to indicate that incorrect parameters were passed to the request.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>401 Unauthorized</td><td>It is used when the client did not submit the appropriate credentials.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>1</td><td>403 Forbidden</td><td>The operation is not allowed given the current status of the resource.
More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>404 Not Found</td><td>It is used when a client provided a URI that cannot be mapped to a valid resource URI.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>406 Not Acceptable</td><td>It is used to indicate that the server cannot provide the any of the content formats supported by the client.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>409 Conflict</td><td>The operation cannot be executed currently, due to a conflict with the state of the resource.
Typically, this is because the application instance resource is in NOT_INSTANTIATED state.
The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>429 Too Many Requests</td><td>It is used when a rate limiter has triggered.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
</table>

7.4.7.3.2 GET

Not supported.

7.4.7.3.3 PUT

Not supported.

7.4.7.3.4 DELETE

Not supported.

7.4.7.3.5 PATCH

Not supported.

7.4.8 Resource: operate application instance task

7.4.8.1 Description

This resource represents the task of changing the operational state of the application instance. The client can use this resource to start or stop an application instance.

7.4.8.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/app_instances/{appInstanceId}/operate

Resource URI variables for this resource are defined in table 7.4.8.2-1.

Table 7.4.8.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2.
appInstanceId	The identifier of the application instance. See note.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request creating the new application instance resource. It can also be retrieved from the "id" attribute in the payload body of that response.	

7.4.8.3 Resource methods

7.4.8.3.1 POST

The POST method is used to change the operational state, i.e. start or stop, of the application instance.

This method shall comply with the URI request and response data structures, and response codes, as specified in tables 7.4.8.3.1-1 and 7.4.8.3.1-2.

Table 7.4.8.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.8.3.1-2: Data structures supported by POST request/response on this resource

	Data type	Cardinality	Remarks	
Request body	OperateAppRequest	1	The payload body in the request contains the information necessary to change the operational state of application instance, as described in clause 6.2.2.8.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	n/a		202 Accepted	The request was accepted for processing, but it is possible that the processing is not yet completed. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created "application LCM operation occurrence" resource that corresponds to this application instance instantiation operation.
	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 the appropriate 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	409 Conflict	The operation cannot be executed currently, due to a conflict with the state of the resource. Typically, this is because the application instance resource is not in INSTANTIATED state. The response body shall contain a ProblemDetails structure, in which the "detail" attribute shall convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.8.3.2 GET

Not supported.

7.4.8.3.3 PUT

Not supported.

7.4.8.3.4 DELETE

Not supported.

7.4.8.3.5 PATCH

Not supported.

7.4.9 Resource: application LCM operation occurrences

7.4.9.1 Description

This resource represents application instance LCM operation occurrences. The client can use this resource to query status information about multiple application instance lifecycle management operation occurrences.

7.4.9.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/app_lcm_op_occs.

Resource URI variables for this resource are defined in table 7.4.9.2-1.

Table 7.4.9.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See Clause 7.2

7.4.9.3 Resource methods

7.4.9.3.1 POST

Not supported.

7.4.9.3.2 GET

The GET method retrieves information of operation status about multiple application instance lifecycle management operation occurrences.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.4.9.3.2-1 and 7.4.9.3.2-2.

Table 7.4.9.3.2-1: URI query parameters of GET method on the resource

Name	Cardinality	Remarks
filter	0..1	Attribute-based filtering parameters according to ETSI GS MEC 009 [4]. The API producer shall support receiving filtering parameters as part of the URI query string. All attribute names that appear in the AppInstanceLcmOpOcc and in data types referenced from it shall be supported in attribute-based filtering parameters. See clause 6.19 in ETSI GS MEC 009 [4] for details.
all_fields	0..1	Include all complex attributes in the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer shall support this parameter.
fields	0..1	Complex attributes of AppInstanceLcmOpOcc to be included into the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer should support this parameter.
exclude_fields	0..1	Complex attributes of AppInstanceLcmOpOcc to be excluded from the response. See clause 6.18 in ETSI GS MEC 009 [4] for details. The API producer should support this parameter.
exclude_default	0..1	Indicates to exclude the following complex attributes of AppInstanceLcmOpOcc from the response. The following attributes shall be excluded from the AppInstanceLcmOpOcc structure in the response body if this parameter is provided, or none of the parameters "all_fields", "fields", "exclude_fields", "exclude_default" are provided: • operationParams; • _links.

Table 7.4.9.3.2-2: Data structures supported by GET request/response on this resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>n/a</td><td></td><td colspan="2"></td></tr>
<tr><th rowspan="6">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>AppInstanceLcmOpOcc</td><td>0..N</td><td>200 OK</td><td>Status information for zero or more application instance lifecycle management operation occurrences was queried successfully.
The response body shall contain in an array the status information about zero or more application lifecycle operation occurrences.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>400 Bad Request</td><td>It is used to indicate that incorrect parameters were passed to the request.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>401 Unauthorized</td><td>It is used when the client did not submit credentials.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
<tr><td>ProblemDetails</td><td>1</td><td>403 Forbidden</td><td>The operation is not allowed given the current status of the resource.
More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.</td></tr>
<tr><td>ProblemDetails</td><td>0..1</td><td>404 Not Found</td><td>It is used when a client provided a URI that cannot be mapped to a valid resource URI.
In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</td></tr>
</table>

	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.

7.4.9.3.3 PUT

Not supported.

7.4.9.3.4 DELETE

Not supported.

7.4.9.3.5 PATCH

Not supported.

7.4.10 Resource: individual application LCM operation occurrence

7.4.10.1 Description

This method represents an individual application LCM operation occurrence. The client can use this resource to read status information about an individual application lifecycle management operation occurrence.

7.4.10.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/app_lcm/v1/app_lcm_op_occs/{appLcmOpOccId}.

Resource URI Variables for this resource are defined in table 7.4.10.2-1.

Table 7.4.10.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2.
appLcmOpOccId	Identifies an individual application LCM operation occurrence.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering an application instance LCM operation. It can also be retrieved from the "appLcmOpOccId" attribute in the AppInstanceLcmOperationOccurrenceNotification.	

7.4.10.3 Resource methods

7.4.10.3.1 POST

Not supported.

7.4.10.3.2 GET

The GET method reads the status information of an individual application LCM operation occurrence, which is used by the procedure of "query application lifecycle operation status" as described in clause 6.3.1.6.

The method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in the tables 7.4.10.3.2-1 and 7.4.10.3.2-2.

Table 7.4.10.3.2-1: URI query parameters of GET method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.4.10.3.2-2: Data structures supported by GET request/response on this resource

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

7.4.10.3.3 PUT

Not supported.

7.4.10.3.4 DELETE

Not supported.

7.4.10.3.5 PATCH

Not supported.

7.5 Resources of granting on Mm3

7.5.1 Resource: grants

7.5.1.1 Description

This resource represents grants. The client can use this resource to obtain permission from the MEO to perform a particular application lifecycle operation.

7.5.1.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/granting/v1/grants.

Resource URI variables for this resource are defined in table 7.5.1.2-1.

Table 7.5.1.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2.

7.5.1.3 Resource methods

7.5.1.3.1 POST

The POST method requests a grant for a particular application lifecycle operation.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in tables 7.5.1.3.1-1 and 7.5.1.3.1-2.

As the result of successfully processing this request, a new "individual grant" resource shall be created. In the synchronous case which is indicated by responding with "201 Created", that resource shall be created before the 200 OK response is returned. In the asynchronous case which is indicated by responding with "202 Accepted", this resource may be created after the response is returned.

Table 7.5.1.3.1-1: URI query parameters of POST method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.5.1.3.1-2: Data structures supported by POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	GrantRequest	1	The POST method is to request a grand for an application lifecycle operation.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	Grant	1	201 Created	Shall be returned when the grant was created successfully (synchronous mode). A representation of the created "Individual grant" resource shall be returned in the response body. The HTTP response shall include a "Location" HTTP header that indicates the URI of the "Individual grant" resource just created.
	n/a		202 Accepted	Shall be returned when the request was accepted for processing, and it is expected to take some time to create the grant (asynchronous mode). The HTTP response shall include a "Location" HTTP header that indicates the URI of the "Individual grant" resource just created.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.5.1.3.2 GET

This method is not supported.

7.5.1.3.3 PUT

Not supported.

7.5.1.3.4 DELETE

Not supported.

7.5.1.3.5 PATCH

Not supported.

7.5.2 Resource: individual grant

7.5.2.1 Description

This resource represents an individual grant. The client can use this resource to read the grant.

It is determined by means outside the scope of the present document, such as configuration or policy, how long an individual grant is available.

7.5.2.2 Resource definition

The possible resource URIs are:

- Resource URI: {apiRoot}/granting/v1/grants/{grantId}.

Resource URI Variables for this resource are defined in table 7.5.2.2-1.

Table 7.5.2.2-1: Resource URI Variables for the resource

Name	Definition
apiRoot	See clause 7.2.
grantId	Identifier of the individual grant.
NOTE: This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request granting a new application lifecycle operation. It can also be retrieved from the "id" attribute in the payload body of that response.	

7.5.2.3 Resource methods

7.5.2.3.1 POST

Not supported.

7.5.2.3.2 GET

The GET method reads a grant.

This method shall comply with the URI query parameters, request and response data structures, and response codes, as specified in the tables 7.5.2.3.2-1 and 7.5.2.3.2-2.

Table 7.5.2.3.2-1: URI query parameters of GET method on the resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.5.2.3.2-2: Data structures supported by GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	N/A	0		
Response body	Data type	Cardinality	Response codes	Remarks
	Grant	1	200 OK	Shall be returned when the grant was read successfully. A representation of the "individual grant" resource shall be returned in the response body.
	n/a		202 Accepted	Shall be returned when the process of creating the grant is ongoing, no grant is available yet. The response body shall be empty. The HTTP response shall include a "Location" HTTP header that contains the URI of the newly-created resource.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.5.2.3.3 PUT

Not supported.

7.5.2.3.4 DELETE

Not supported.

7.5.2.3.5 PATCH

Not supported.

7.6 Resources of MEPM's application lifecycle management on Mm3

Refer to clause 7.4.

Annex A (informative): Application package state model

A.1 Introduction

This annex presents the state model of application package. The steps before onboarding an application package is out of scope of this model.

A.2 State model

This clause describes the state model of application package in the MEO. It consists of the onboarding phase and onboarded phase.

Figure A.2-1 shows the state model of application package.

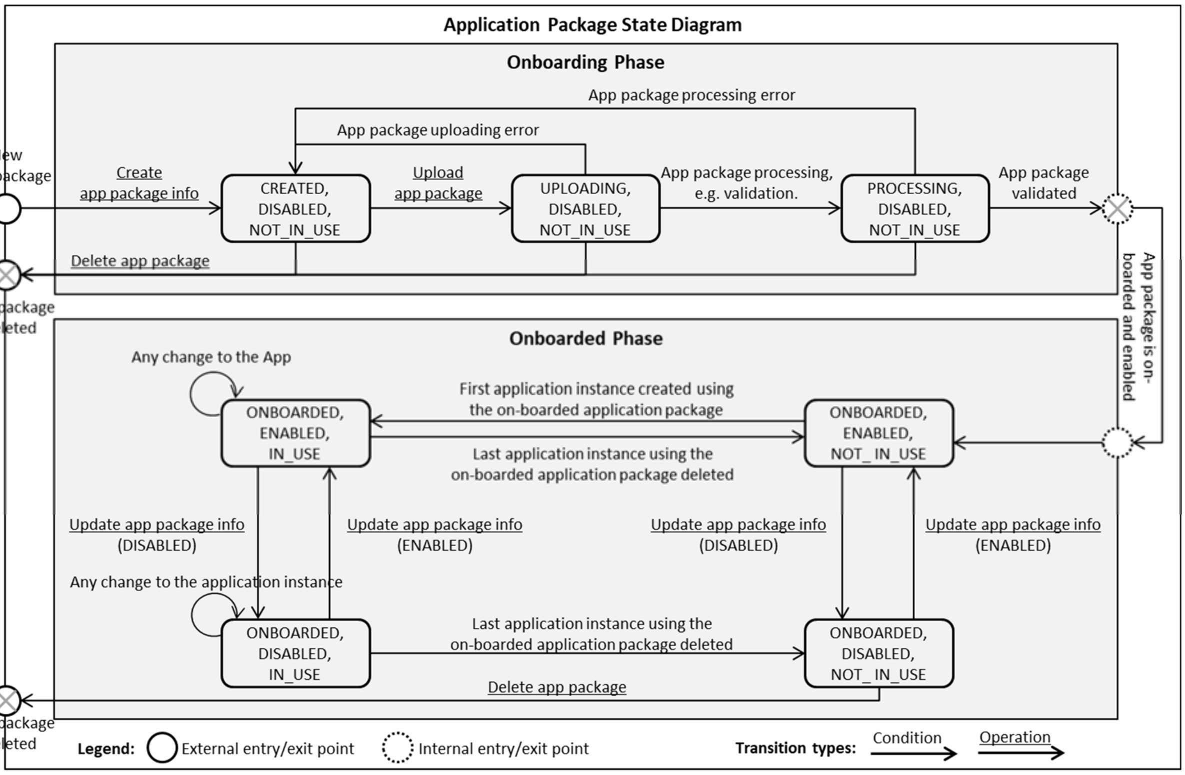
Application Package State Diagram
Onboarding Phase
App package processing error
App package uploading error
New App package
Create app package info
CREATED, DISABLED, NOT_IN_USE
Upload app package
UPLOADING, DISABLED, NOT_IN_USE
App package processing, e.g. validation.
PROCESSING, DISABLED, NOT_IN_USE
App package validated
Delete app package
App package deleted
App package is on-boarded and enabled
Onboarded Phase
Any change to the App
ONBOARDED, ENABLED, IN_USE
First application instance created using the on-boarded application package
ONBOARDED, ENABLED, NOT_IN_USE
Last application instance using the on-boarded application package deleted
Update app package info (DISABLED)
Update app package info (ENABLED)
Update app package info (DISABLED)
Update app package info (ENABLED)
Any change to the application instance
ONBOARDED, DISABLED, IN_USE
Last application instance using the on-boarded application package deleted
ONBOARDED, DISABLED, NOT_ IN_USE
Delete app package
App package deleted
Legend:
External entry/exit point
Internal entry/exit point
Transition types:
Condition
Operation

Figure A.2-1: Application package state model

The onboarding state, operational state and usage state are represented by the "onboardingState", "operationalState" and "usageState" attributes respectively in the "AppPkgInfo" information element specified in table 6.2.3.3.2-1.

Annex B (informative): Bibliography

- ETSI GS MEC 003: "Multi-access Edge Computing (MEC); Framework and Reference Architecture".
- ETSI GS NFV-IFA 013: "Network Functions Virtualisation (NFV); Management and Orchestration; Os-Ma-Nfvo reference point - Interface and Information Model Specification".

History

Document history		
V1.1.1	July 2017	Publication
V2.1.1	November 2019	Publication