



GROUP SPECIFICATION

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

Disclaimer

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

Reference

DGS/MEC-0010-2AppLcRulesReqMgm

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 only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

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

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2017.
All rights reserved.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.
3GPP™ and **LTE™** are Trade Marks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

oneM2M logo is protected for the benefit of its Members
GSM® and the GSM logo are Trade Marks registered and owned by the GSM Association.

Contents

Intellectual Property Rights	7
Foreword.....	7
Modal verbs terminology.....	7
1 Scope	8
2 References	8
2.1 Normative references	8
2.2 Informative references.....	8
3 Definitions and abbreviations.....	9
3.1 Definitions.....	9
3.2 Abbreviations	9
4 Specification level requirements	9
4.1 Requirements.....	9
4.1.1 Requirements for reference point Mm1	9
4.1.1.1 General requirements	9
4.1.1.2 Interface requirements.....	10
4.1.1.2.1 Application package management interface requirements	10
4.1.1.2.2 Application lifecycle management interface requirements.....	10
4.1.2 Requirements for reference point Mm3	10
4.1.2.1 General requirements	10
4.1.2.2 Interface requirements.....	11
4.1.2.2.1 Application package management interface requirements	11
4.1.2.2.2 Application lifecycle management interface requirements.....	11
4.1.2.2.3 Application lifecycle change notification interface requirements	11
4.1.3 Requirements for application package.....	12
4.1.3.1 General requirements	12
4.1.3.2 Application descriptor requirements	13
5 Message flows to support application lifecycle management	13
5.1 General	13
5.2 Application package on-boarding.....	13
5.2.1 General.....	13
5.2.2 On-board application package	14
5.2.3 Query application package.....	14
5.2.4 Disable application package	15
5.2.5 Enable application package.....	15
5.2.6 Delete application package	16
5.3 Application instantiation	17
5.4 Application termination.....	18
6 Information models and interfaces	19
6.1 Applicable reference points	19
6.2 Information models	19
6.2.1 Application descriptor information model	19
6.2.1.1 Introduction.....	19
6.2.1.2 Type: Application descriptor.....	19
6.2.1.2.1 Description	19
6.2.1.2.2 Attributes	19
6.2.1.3 Type: VirtualComputeDescription.....	21
6.2.1.3.1 Description	21
6.2.1.3.2 Attributes	21
6.2.1.4 Type: SwImageDescriptor	21
6.2.1.4.1 Description	21
6.2.1.4.2 Attributes	21
6.2.1.5 Type: VirtualStorageDescriptor	21
6.2.1.5.1 Description	21

6.2.1.5.2	Attributes	21
6.2.1.6	Type: AppExternalCpd	21
6.2.1.6.1	Description	21
6.2.1.6.2	Attributes	22
6.2.1.7	Type: ServiceDescriptor	22
6.2.1.7.1	Description	22
6.2.1.7.2	Attributes	22
6.2.1.8	Type: FeatureDependency	22
6.2.1.8.1	Description	22
6.2.1.8.2	Attributes	22
6.2.1.9	Type: TrafficRuleDescriptor	23
6.2.1.9.1	Description	23
6.2.1.9.2	Attributes	23
6.2.1.10	Type: TrafficFilter	23
6.2.1.10.1	Description	23
6.2.1.10.2	Attributes	23
6.2.1.11	Type: InterfaceDescriptor	24
6.2.1.11.1	Description	24
6.2.1.11.2	Attributes	24
6.2.1.12	Type: TunnelInfo	24
6.2.1.12.1	Description	24
6.2.1.12.2	Attributes	24
6.2.1.13	Type: DnsRuleDescriptor	25
6.2.1.13.1	Description	25
6.2.1.13.2	Attributes	25
6.2.1.14	Type: LatencyDescriptor	25
6.2.1.14.1	Description	25
6.2.1.14.2	Attributes	25
6.2.1.15	Type: TerminateAppInstanceOpConfig	25
6.2.1.15.1	Description	25
6.2.1.15.2	Attributes	25
6.2.1.16	Type: ChangeAppInstanceStateOpConfig	26
6.2.1.16.1	Description	26
6.2.1.16.2	Attributes	26
6.2.1.17	Type: ServiceDependency	26
6.2.1.17.1	Description	26
6.2.1.17.2	Attributes	26
6.2.1.18	Type: TransportDependency	26
6.2.1.18.1	Description	26
6.2.1.18.2	Attributes	26
6.2.1.19	Type: TransportDescriptor	27
6.2.1.19.1	Description	27
6.2.1.19.2	Attributes	27
6.3	Interfaces	28
6.3.1	Application lifecycle management interface	28
6.3.1.1	Description	28
6.3.1.2	Create application instance identifier operation	28
6.3.1.2.1	Description	28
6.3.1.2.2	Input parameters	29
6.3.1.2.3	Output parameters	29
6.3.1.2.4	Operation results	29
6.3.1.3	Application instantiation operation	29
6.3.1.3.1	Definition	29
6.3.1.3.2	Input parameters	29
6.3.1.3.3	Output parameters	30
6.3.1.3.4	Operation results	30
6.3.1.4	Change application instance operational state operation	30
6.3.1.4.1	Description	30
6.3.1.4.2	Input parameters	31
6.3.1.4.3	Output parameters	32
6.3.1.4.4	Operation results	32
6.3.1.5	Query application instance information operation	33

6.3.1.5.1	Description	33
6.3.1.5.2	Input parameters	33
6.3.1.5.3	Output parameters	33
6.3.1.5.4	Operation results.....	33
6.3.1.6	Query application lifecycle operation status	34
6.3.1.6.1	Description	34
6.3.1.6.2	Input parameters	34
6.3.1.6.3	Output parameters	34
6.3.1.6.4	Operation results.....	34
6.3.1.7	Application instance terminate operation.....	34
6.3.1.7.1	Description	34
6.3.1.7.2	Input parameters	35
6.3.1.7.3	Output parameters	36
6.3.1.7.4	Operation results.....	36
6.3.1.8	Delete application instance identifier operation	36
6.3.1.8.1	Description	36
6.3.1.8.2	Input parameters	36
6.3.1.8.3	Output parameters	36
6.3.1.8.4	Operation results.....	36
6.3.2	Application lifecycle change notification interface	37
6.3.2.1	Description	37
6.3.2.2	Subscribe operation.....	37
6.3.2.2.1	Description	37
6.3.2.2.2	Input parameters	37
6.3.2.2.3	Output parameters	37
6.3.2.2.4	Operation results.....	37
6.3.2.3	Notify operation	38
6.3.2.3.1	Description	38
6.3.3	Application package management interface	38
6.3.3.1	Fetch on-boarded application package operation	38
6.3.3.1.1	Definition.....	38
6.3.3.1.2	Input parameters	38
6.3.3.1.3	Output parameters	39
6.3.3.1.4	Operation results.....	39
6.3.3.2	Query application package operation	39
6.3.3.2.1	Definition.....	39
6.3.3.2.2	Input parameters	39
6.3.3.2.3	Output parameters	39
6.3.3.2.4	Operation results.....	40
6.3.3.3	Subscribe operation.....	40
6.3.3.3.1	Definition.....	40
6.3.3.3.2	Input parameters	40
6.3.3.3.3	Output parameters	40
6.3.3.3.4	Operation results.....	40
6.3.3.4	Notify Application Package operation	40
6.3.3.4.1	Definition.....	40
6.3.3.5	Onboarding operation.....	41
6.3.3.5.1	Definition.....	41
6.3.3.5.2	Input parameters	41
6.3.3.5.3	Output parameters	41
6.3.3.5.4	Operation results.....	42
6.3.3.6	Enable operation	42
6.3.3.6.1	Definition.....	42
6.3.3.6.2	Input parameters	42
6.3.3.6.3	Output parameters	42
6.3.3.6.4	Operation results.....	42
6.3.3.7	Disable operation	42
6.3.3.7.1	Definition.....	42
6.3.3.7.2	Input parameters	43
6.3.3.7.3	Output parameters	43
6.3.3.7.4	Operation results.....	43
6.3.3.8	Query operation.....	43

6.3.3.8.1	Definition.....	43
6.3.3.8.2	Input parameters	43
6.3.3.8.3	Output parameters	44
6.3.3.8.4	Operation results.....	44
6.3.3.9	Delete operation	44
6.3.3.9.1	Definition.....	44
6.3.3.9.2	Input parameters	44
6.3.3.9.3	Output parameters	44
6.3.3.9.4	Operation results.....	44
6.3.3.10	Abort application package deletion operation.....	45
6.3.3.10.1	Description	45
6.3.3.10.2	Input parameters	45
6.3.3.10.3	Output parameters	45
6.3.3.10.4	Operation results.....	45
Annex A (normative):	Application package state model	46
A.1	Introduction	46
A.2	State model.....	46
Annex B (informative):	Bibliography.....	47
History		48

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<https://ipr.etsi.org>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Mobile Edge Computing (MEC).

The present document is part 2 of a multi-part deliverable covering Mobile Edge 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 mobile edge host, and describes interfaces over the reference points to support application lifecycle management. It also describes application rules and requirements, application-related events and mobility handling. 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 (V2.1.1): "Network Functions Virtualisation (NFV); Management and Orchestration; VNF Packaging Specification".

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: "Mobile Edge Computing (MEC); Terminology".
- [i.2] ETSI GS MEC 002: "Mobile Edge Computing (MEC); Technical Requirements".
- [i.3] ETSI GS NFV-IFA 007 (V2.1.1): "Network Functions Virtualisation (NFV); Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification".
- [i.4] ETSI GS MEC 011: "Mobile Edge Computing (MEC); Mobile Edge Platform Application Enablement".
- [i.5] ETSI GS NFV 003 (V1.2.1): "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".
- [i.6] ETSI GS MEC 009: "Mobile Edge Computing (MEC); General principles for Mobile Edge Service APIs".

3 Definitions and abbreviations

3.1 Definitions

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

application descriptor: descriptor provided by the application provider, and describes the application rules and requirements of a mobile edge application

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

3.2 Abbreviations

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

CPD	Connection Point Descriptor
DNS	Domain Name System
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
IP	Internet Protocol
MAC	Media Access Control
ME	Mobile Edge
MEO	Mobile Edge Orchestrator
MEPM	Mobile Edge Platform Manager
QCI	QoS Class Identifier
RNIS	Radio Network Information Service
URL	Uniform Resource Locator

4 Specification level requirements

4.1 Requirements

4.1.1 Requirements for reference point Mm1

4.1.1.1 General requirements

The Mm1 reference point between the Mobile Edge Orchestrator and the OSS is used for on-boarding application packages, triggering the instantiation and the termination of mobile edge applications in the mobile edge system. Table 4.1.1.1-1 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 4.1.1.1-1: Mm1 reference point requirements

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

4.1.1.2 Interface requirements

4.1.1.2.1 Application package management interface requirements

Table 4.1.1.2.1-1 specifies requirements applicable to the application package management interface produced by the Mobile Edge Orchestrator on the Mm1 reference point. Those requirements are derived from clause 5.2 of ETSI GS MEC 002 [i.2].

Table 4.1.1.2.1-1: 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.
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.

4.1.1.2.2 Application lifecycle management interface requirements

Table 4.1.1.2.2-1 specifies requirements applicable to the application lifecycle management interface produced by the Mobile Edge Orchestrator on the Mm1 reference point.

Table 4.1.1.2.2-1: 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).
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.

4.1.2 Requirements for reference point Mm3

4.1.2.1 General requirements

The Mm3 reference point between the Mobile Edge Orchestrator and the Mobile Edge Platform Manager is used for the management of the application lifecycle, application rules and requirements and keeping track of available mobile edge services, etc. Table 4.1.2.1-1 specifies requirements related to application lifecycle management applicable to the Mm3 reference point.

Table 4.1.2.1-1: Mm3 reference point requirements

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

4.1.2.2 Interface requirements

4.1.2.2.1 Application package management interface requirements

Table 4.1.2.2.1-1 specifies requirements applicable to the Application Package Management interface produced by the MEO on the Mm3 reference point.

Table 4.1.2.2.1-1: Application Package Management interface requirements

Numbering	Functional requirement
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.

4.1.2.2.2 Application lifecycle management interface requirements

Table 4.1.2.2.2-1 specifies requirements applicable to the Application Lifecycle Management interface produced by the Mobile Edge Platform Manager on the Mm3 reference point.

Table 4.1.2.2.2-1: Application Lifecycle Management interface requirements

Numbering	Functional requirement
Mm3.AppLcm.001	The Application Lifecycle Management interface produced by the Mobile Edge Platform Manager on the Mm3 reference point shall support instantiating an Application.
Mm3.AppLcm.002	The Application Lifecycle Management interface produced by the Mobile Edge Platform Manager on the Mm3 reference point shall support terminating an application instance.
Mm3.AppLcm.003	The Application Lifecycle Management interface produced by the Mobile Edge 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 Mobile Edge 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 Mobile Edge 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.

4.1.2.2.3 Application lifecycle change notification interface requirements

Table 4.1.2.2.3-1 specifies requirements applicable to the Application Lifecycle Change Notification interface produced by the Mobile Edge Platform manager on the Mm3 reference point.

Table 4.1.2.2.3-1: Application Lifecycle Change Notification interface requirements

Numbering	Functional requirement
Mm3.AppLccn.001	The Application Lifecycle Change Notification interface produced by the Mobile Edge 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 Mobile Edge 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 Mobile Edge Platform Manager on the Mm3 reference point shall support indicating the start of the lifecycle procedure, the end and the results of the lifecycle procedure including any error produced from the lifecycle procedure.
Mm3.AppLccn.004	The Application Lifecycle Change Notification interface produced by the Mobile Edge 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.

4.1.3 Requirements for application package

4.1.3.1 General requirements

Table 4.1.3.1-1 specifies requirements related to application lifecycle management applicable to the application package.

Table 4.1.3.1-1: 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 notes 1 and 2.
AppPkt.002	The application package shall contain an application descriptor that describes the application requirements and rules which are required or preferred by the mobile edge application. See note 1.
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.
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 1.
NOTE 1: The application package, the software image(s), the manifest file and the application descriptor are provided by the application provider.	
NOTE 2: In the present document it is assumed that the software image is a VM image.	

4.1.3.2 Application descriptor requirements

Table 4.1.3.2-1 specifies requirements related to application lifecycle management applicable to the application descriptor.

Table 4.1.3.2-1: 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 mobile edge application requires to run.
AppDesc.005	The application descriptor shall support describing a list of additional services that a mobile edge application may use if available.
AppDesc.006	The application descriptor shall support describing a list of features a mobile edge application requires to run.
AppDesc.007	The application descriptor shall support describing a list of additional features a mobile edge application can may 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 ME system (e.g. for redirection of traffic to local host).
AppDesc.010	The application descriptor shall support a description of latency required by the mobile edge application.

5 Message flows to support application lifecycle management

5.1 General

Message flows in this clause are informative.

5.2 Application package on-boarding

5.2.1 General

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

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

5.2.2 On-board application package

The message flow of on-board application package is used to make application package available to the mobile edge 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.

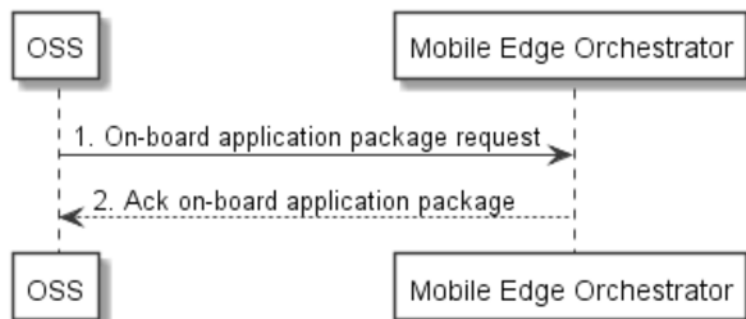


Figure 5.2.2-1: On-board application package

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

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

The Mobile Edge Orchestrator allocates a unique application package ID for the on-boarded mobile edge application package and related status information, and keeps a record of on-boarded application packages. Optionally the Mobile Edge 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 Mobile Edge Orchestrator notifies the subscribers to AppPackageOnBoardingNotification of the on-boarding of the mobile edge application package.

- 2) The Mobile Edge Orchestrator acknowledges the application package on-boarding to the OSS. The application package is now available in the mobile edge system.

5.2.3 Query application package

Query application package allows returning the information (such as version, application rules and requirements, etc.) of an application package(s) to the OSS. The detailed description of the flow is in figure 5.2.3-1.

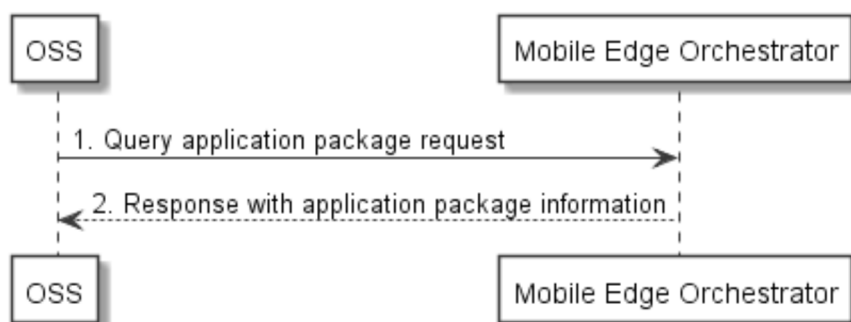


Figure 5.2.3-1: Query application package

- 1) The OSS sends a query application package request (including a filter) to the Mobile Edge Orchestrator to query information of application package(s).

The Mobile Edge Orchestrator authorizes the request.

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

5.2.4 Disable application package

Disabling application package refers to the process of marking an application package as disabled in the mobile edge 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.

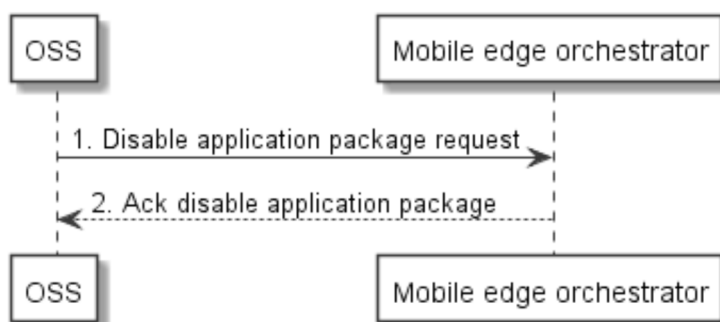


Figure 5.2.4-1: Disable application package

- 1) The OSS sends a request to disable application package to the Mobile Edge Orchestrator.

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

- 2) The Mobile Edge Orchestrator acknowledges the disable application package request.

5.2.5 Enable application package

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

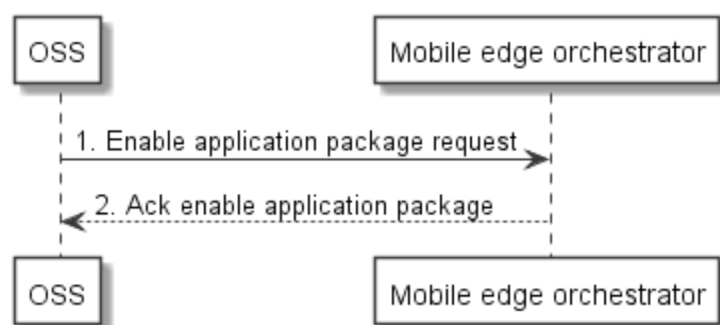


Figure 5.2.5-1 Enable application package

- 1) The OSS sends an enable application package request to the Mobile Edge Orchestrator.

The Mobile Edge Orchestrator processes the request and checks if the application package exists, is disabled and is not marked as deletion pending. If the application package is disabled and is not marked as deletion pending, the Mobile Edge Orchestrator marks the application package as enabled in the mobile edge system.

- 2) The Mobile Edge Orchestrator acknowledges the application package enable request.

5.2.6 Delete application package

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

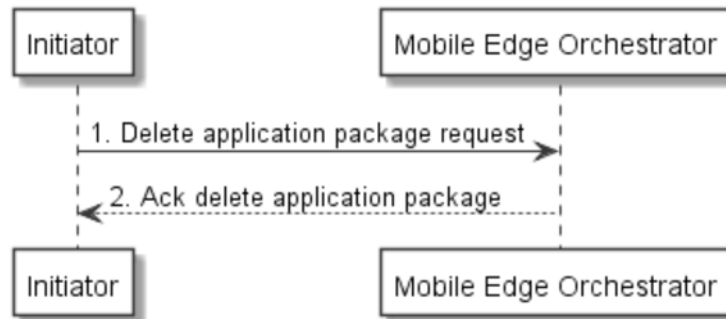


Figure 5.2.6-1: Delete application package

- 1) The OSS sends a delete application package request to the Mobile Edge Orchestrator.

The Mobile Edge Orchestrator checks whether the application package is disabled and whether the application package is in use. If the application package is disabled but is in use, the Mobile Edge Orchestrator set the state of the application package as deletion pending. If the application package is disabled and is not in use, the Mobile Edge Orchestrator removes the application package from the mobile edge system.

If the application package is in deletion pending, the Mobile Edge Orchestrator will check if all application instances of the application package have been terminated later. If there is no application instance using the application package any more, the Mobile Edge Orchestrator removes the application package from the mobile edge system.

- 2) The Mobile Edge Orchestrator acknowledges application package deletion request.

5.3 Application instantiation

The message flow of application instantiation is used to instantiate an application instance in the mobile edge system. The detailed description of the flow is in figure 5.3-1.

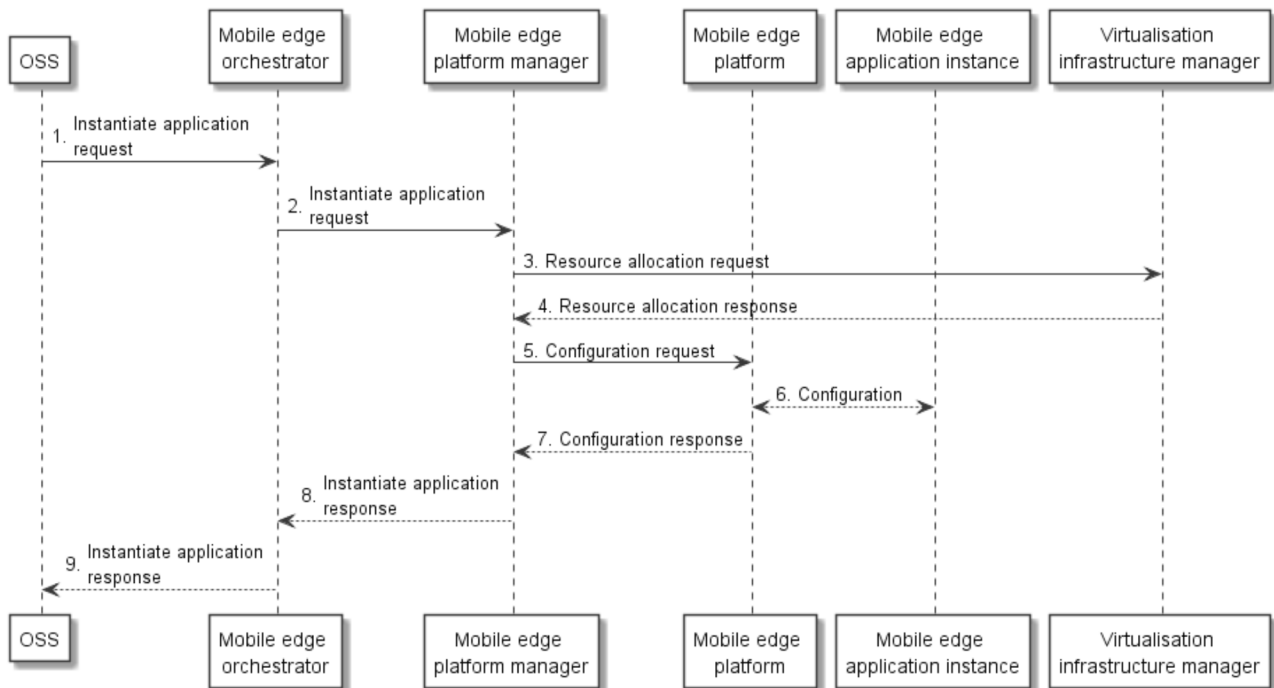


Figure 5.3-1: Application instantiation flow

- 1) The OSS sends an instantiate application request to the Mobile Edge Orchestrator.
- 2) Mobile Edge Orchestrator checks the application instance configuration data, and authorizes the request. The Mobile Mdge Orchestrator selects the mobile edge host (and corresponding Mobile Edge Platform Manager), and sends an instantiate application request to the Mobile Edge Platform Manager.
- 3) The Mobile Edge Platform Manager sends a resource allocation request to the virtualization infrastructure manager, with the requested resource including compute, storage, and network resources. The mobile edge platform will include application image information (e.g. a link to the image or an ID of the application image) in the request.
- 4) The virtualization infrastructure manager allocates the resources according to the request of the Mobile Edge Platform Manager. And if the application image is available, the virtualization infrastructure manager loads the virtual machine with the application image, and runs the VM and the application instance. The virtualization infrastructure manager sends resource allocation response to the Mobile Edge Platform Manager.
- 5) The Mobile Edge Platform Manager sends configuration request to the mobile edge platform. In this message, the Mobile Edge 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 1: This step is only for the completeness of the flow, and will not be specified in the present document.

- 6) The mobile edge platform configures the Traffic rules and DNS rules for the application instance. The mobile edge 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. For such purpose, the mobile edge platform needs to communicate with the application instance regarding to its state via Mp1 interface if it is supported by the mobile edge application. After the application instance runs normally, the mobile edge platform provides the available service information to the application. The details of how the mobile edge application instance interacts with mobile edge platform and how the mobile edge platform handles services can be found in ETSI GS MEC 011 [i.4].

- 7) The mobile edge platform sends a configuration response to the Mobile Edge Platform Manager.

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

- 8) The Mobile Edge Platform Manager sends an instantiate application response to the Mobile Edge Orchestrator. The Mobile Edge Platform Manager includes the information of the resources allocated to the application instance to the Mobile Edge Orchestrator.
- 9) The Mobile Edge Orchestrator sends an instantiate application response to the OSS, returning the results of the instantiation procedure. The Mobile Edge Orchestrator also returns the application instance ID to the OSS if the flow is successful.

5.4 Application termination

The message flow of application instance termination is used to terminate an application instance in the mobile edge system. The detailed description of the flow is in figure 5.4-1.

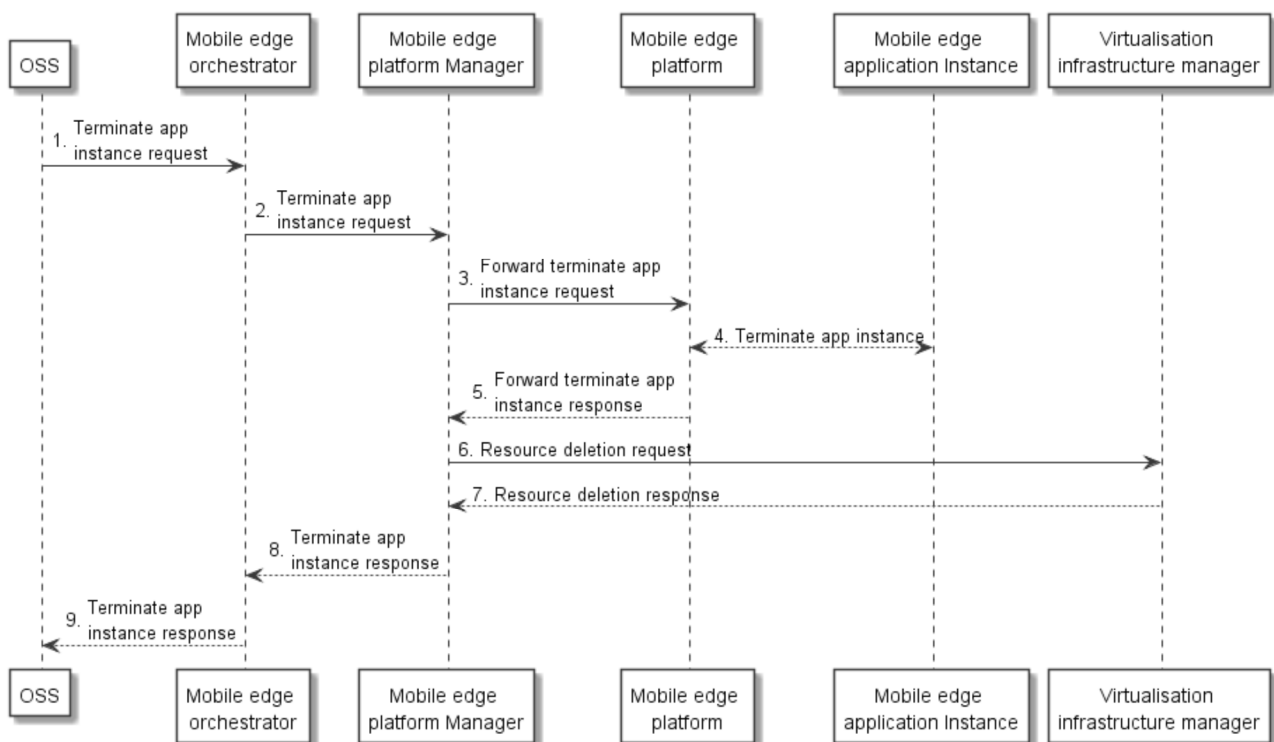


Figure 5.4-1: Instance Termination information flow

- 1) The OSS sends a terminate application instance request to the Mobile Edge Orchestrator. The message includes the mobile edge application instance ID to be terminated.
- 2) The Mobile Edge Orchestrator authorizes the request, and verifies the existence of requested instance(s). The Mobile Edge Orchestrator sends a terminate application instance request to the Mobile Edge Platform Manager that manages the mobile edge application instance(s) to be terminated.
- 3) The Mobile Edge Platform Manager sends a forward terminate application instance request to the mobile edge 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 mobile edge application, and graceful termination is requested, the mobile edge platform notifies the mobile edge application instance of the termination event. The mobile edge application instance executes the actions needed before it's been terminated by the mobile edge platform, the actual action(s) the mobile edge application instance will perform for the application level termination is up to mobile edge application, and is out of scope of the present document. After the mobile edge application instance finishes application level termination, it will inform the mobile edge platform that it is ready to be terminated. The mobile edge platform may set a timer for the application level termination. After the timer expires, the mobile edge platform will shut down the application regardless of the progress of application level termination.
- 5) The mobile edge platform sends a forward terminate application instance response to the Mobile Edge 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 Mobile Edge Platform Manager sends a resource deletion request to the corresponding virtualization infrastructure manager, to terminate the virtual machine and release the resources.
- 7) The virtualization infrastructure manager releases the resources allocated for the application instance to be terminated. And the virtualization infrastructure manager sends resource a deletion response to the Mobile Edge Platform Manager.
- 8) The Mobile Edge Platform Manager sends a terminate application instance response to the Mobile Edge Orchestrator.
- 9) The Mobile Edge Orchestrator sends a terminate application instance response to the OSS.

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: Application descriptor

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	Qualifier	Cardinality	Data type	Description
appDId	M	1	String	Identifier of this mobile edge application descriptor. This attribute shall be globally unique. See note 1.
appName	M	1	String	Name to identify the mobile edge application.
appProvider	M	1	String	Provider of the application and of the AppD.
appSoftVersion	M	1	String	Identifies the version of software of the mobile edge application.
appDVersion	M	1	String	Identifies the version of the application descriptor.
mecVersion	M	1..N	String	Identifies version(s) of ME system compatible with the mobile edge application described in this version of the AppD.
applInfoName	M	0..1	String	Human readable name for the ME application product. May change during the ME application product lifetime.
appDescription	M	1	String	Human readable description of the mobile edge application.
virtualComputeDescriptor	M	1	VirtualComputeDescriptor	Describes CPU, Memory and acceleration requirements of the Virtualisation machine.
swImageDescriptor	M	1	SwImageDescriptor	Describes the software image which is directly loaded on the virtualisation machine instantiating this Application.
virtualStorageDescriptor	M	0..N	VirtualStorageDescriptor	Defines descriptors of virtual storage resources to be used by the mobile edge application.
appExtCpd	M	0..N	AppExternalCpd	Describes external interface(s) exposed by this mobile edge application.
appServiceRequired	M	0..N	ServiceDependency	Describes services a mobile edge application requires to run.
appServiceOptional	M	0..N	ServiceDependency	Describes services a mobile edge application may use if available.
appServiceProduced	M	0..N	ServiceDescriptor	Describes services a mobile edge application is able to produce to the platform or other mobile edge applications. Only relevant for service-producing apps.
appFeatureRequired	M	0..N	FeatureDependency	Describes features a mobile edge application requires to run.
appFeatureOptional	M	0..N	FeatureDependency	Describes features a mobile edge application may use if available.
transportDependencies	M	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	M	0..N	TrafficRuleDescriptor	Describes traffic rules the mobile edge application requires.
appDNSRule	M	0..N	DNSRuleDescriptor	Describes DNS rules the mobile edge application requires.
appLatency	M	0..1	LatencyDescriptor	Describes the maximum latency tolerated by the mobile edge application.
terminateAppInstanceOpConfig	M	0..1	TerminateAppInstanceOpConfig	Configuration parameters for the Terminate application instance operation.
changeAppInstanceStateOpConfig	M	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 ME 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: VirtualComputeDescription

6.2.1.3.1 Description

The VirtualComputeDescription 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 ME 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 ME application.

6.2.1.4 Type: SwImageDescriptor

6.2.1.4.1 Description

The SwImageDescriptor data type describes a software image of a ME 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 ME Application, the VNFD refers to AppD in MEC, and the VDU refers to ME application.

6.2.1.5 Type: VirtualStorageDescriptor

6.2.1.5.1 Description

The VirtualStorageDescriptor data type describes a virtual storage required by a ME 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 ME Application, the VNFD refers to AppD in MEC, and the VDU refers to ME 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 ME 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	Qualifier	Cardinality	Data type	Description
virtualNetworkInterfaceRequirements	M	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 ME service produced by a service-providing ME 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	Qualifier	Cardinality	Data type	Description
serName	M	1	String	The name of the service, for example, RNIS, LocationService, etc.
serCategory	M	0..1	CategoryRef	A Category reference of the service, defined in ETSI GS MEC 011 [i.4].
version	M	1	String	The version of the service.
transportsSupported	M	0..N	Structure (inlined)	Indicates transports and serialization formats supported made available to the service-consuming application. Defaults to REST + JSON if absent.
>transport	M	1	TransportDescriptor	Information about the transport in this binding.
>serializers	M	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 ME application related to a feature of ME 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	Qualifier	Cardinality	Data type	Description
featureName	M	1	String	The name of the feature, for example, UserApps, UEIdentity, etc.
version	M	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 ME 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	Qualifier	Cardinality	Data type	Description
trafficRuleId	M	1	String	Identifies the traffic rule.
filterType	M	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	M	1	Int	Priority of this traffic rule. If traffic rule conflicts, the one with higher priority take precedence.
trafficFilter	M	1..N	TrafficFilter	The filter used to identify specific flow/packets that need to be handled by the MEC host.
action	M	1	Enum (inlined)	Identifies the action of the ME 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	M	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 ME 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	Qualifier	Cardinality	Data type	Description
srcAddress	M	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	M	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	M	0..N	String	A port or a range of ports.
dstPort	M	0..N	String	A port or a range of ports.
Protocol	M	0..N	String	Specify the protocol of the traffic filter.
Token	M	0..N	String	Used for token based traffic rule.
srcTunnelAddress	M	0..N	String	Used for GTP tunnel based traffic rule.
tgtTunnelAddress	M	0..N	String	Used for GTP tunnel based traffic rule.
srcTunnelPort	M	0..N	String	Used for GTP tunnel based traffic rule.
dstTunnelPort	M	0..N	String	Used for GTP tunnel based traffic rule.
qCl	M	0..1	Int	Used to match all packets that have the same QCl.
dSCP	M	0..1	Int	Used to match all IPv4 packets that have the same DSCP.
tC	M	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 ME 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	Qualifier	Cardinality	Data type	Description
interfaceType	M	1	Enum (inlined)	Type of interface: TUNNEL, MAC, IP, etc.
tunnelInfo	M	0..1	TunnelInfo	Included only if the destination address type is tunnel.
srcMACAddress	M	0..1	String	If the interface type is MAC, the source address identifies the MAC address of the interface.
dstMACAddress	M	0..1	String	If the interface type is MAC, the destination address identifies the MAC address of the destination. Only used for dstInterface.
dstIPAddress	M	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 ME 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	Qualifier	Cardinality	Data type	Description
tunnelType	M	1	Enum (inlined)	Type of tunnel: GTP-U, GRE, etc.
tunnelDstAddress	M	1	String	Destination address of the tunnel
tunnelSrcAddress	M	1	String	Source address of the tunnel
tunnelSpecificData	M	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 ME 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	Qualifier	Cardinality	Data type	Description
dnsRuleId	M	1	String	Identifies the DNS Rule
domainName	M	1	String	FQDN of the DNS rule
ipAddressType	M	1	Enum (inlined)	Specifies the IP address type, value: IP_V6, IP_V4
ipAddress	M	1	String	IP address given by the DNS rule
ttl	M	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 ME 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	Qualifier	Cardinality	Data type	Description
timeUnit	M	1	UInt32	Time unit, e.g. ms
latency	M	1	String	The value of the latency
NOTE: The meaning of latency will be decided in stage 3 specification.				

6.2.1.15 Type: TerminateAppInstanceOpConfig

6.2.1.15.1 Description

The TerminateAppInstanceOpConfig data type supports the specification of ME 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 ME 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 ME application related to a ME 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	Qualifier	Cardinality	Data type	Description
serName	M	1	String	The name of the service, for example, RNIS, LocationService, etc.
serCategory	M	0..1	CategoryRef	A Category reference of the service.
version	M	1	String	The version of the service.
serTransportDependencies	M	0..N	TransportDependency	Indicates transport and serialization format dependencies of consuming the service. Defaults to REST + JSON if absent. See note.
requestedPermissions	M	0..N	Not specified	Requested permissions regarding the access of the application to the service. See clause 8.2 of ETSI GS MEC 009 [i.6] 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 ME application supports for the consumption of the ME 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 ME 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	Qualifier	Cardinality	Data type	Description
transport	M	1	TransportDescriptor	Information about the transport in this transport binding.
serializers	M	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	M	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	Qualifier	Cardinality	Data type	Description
type	M	1	TransportTypes	Type of the transport, as defined in the TransportTypes type in ETSI GS MEC 011 [i.4]
protocol	M	1	String	The name of the protocol used. Shall be set to "HTTP" for a REST API.
version	M	1	String	The version of the protocol used.
security	M	1	SecurityInfo	Information about the security used by the transport in ETSI GS MEC 011 [i.4].

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.

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 and Query application lifecycle operation status.

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	Qualifier	Cardinality	Data type	Description
appDId	M	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	M	0..1	String	Human-readable name of the application instance to be created.
appInstanceDescription	M	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	Qualifier	Cardinality	Data type	Description
appInstanceDId	M	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 mobile edge 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	Qualifier	Cardinality	Data type	Description
appInstanceId	M	1	String	Identifier of the application instance created by "Create application instance identifier" operation.
virtualComputeDescriptor	M	0..1	VirtualComputeDescriptor	Describes CPU, Memory and acceleration requirements of the Virtualisation machine for the mobile edge application instance to be created. See note 1.
virtualStorageDescriptor	M	0..N	VirtualStorageDescriptor	Defines descriptors of virtual storage resources to be used by the mobile edge application instance to be created. See note 1.
selectedMEHostInfo	M	1..N	MEHostInformation	Describes the information of selected mobile edge host for the application instance. See notes 2 and 3.
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.				

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

6.3.1.3.4 Operation results

In case of success, the mobile edge 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 mobile edge application instance, including starting and stopping the application instance.

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

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

A mobile edge application instance may be in the following states:

- **STARTED:** the mobile edge application instance is up and running.

- STOPPED: the mobile edge application instance has been shut down.

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

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

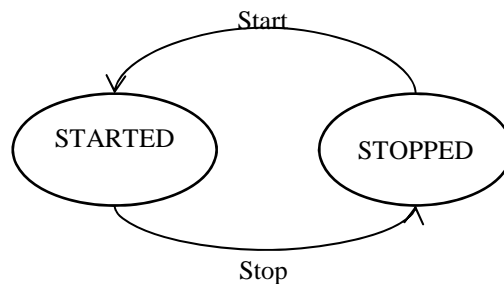


Figure 6.3.1.4.1-1: Change application instance state diagram

It depends on the mobile edge application capabilities, and is declared in the AppD, whether this operation is supported for a particular mobile edge 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
ChangeAppInstanceStateRequest	Mandatory	OSS → MEO MEO → MEPM
ChangeAppInstanceStateResponse	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	Qualifier	Cardinality	Data type	Description
appInstanceId	M	1	String	Identifier of the mobile edge application instance.
changeStateTo	M	1	Enum (inlined)	The desired state to change the VNF to. Permitted values are: START, STOP.
stopType	M	0..1	Enum (inlined)	Signals whether forceful or graceful stop is requested. Allowed values are: FORCEFUL and GRACEFUL. In case of FORCEFUL stop, the mobile edge application is stopped immediately. Note that if the mobile edge 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 mobile edge system gives time to the mobile edge application for application level stop (e.g. via Mp1 interaction). Once this was successful, or after a timeout, the mobile edge system stops the mobile edge application. If the mobile edge application does not support Mp1, the stopType shall be set to FORCEFUL.
gracefulStopTimeout	M	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 mobile edge system waits for the successful application level stop, no matter how long it takes, before stopping the mobile edge 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	Qualifier	Cardinality	Data type	Description
lifecycleOperationOccurrenceId	M	1	String	The identifier of the mobile edge application lifecycle operation occurrence.

6.3.1.4.4 Operation results

In case of success, the state of the mobile edge 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 MEPM and the MEO.

Table 6.3.1.5.1-1: Query application instance information operation

Message	Requirement	Direction
QueryAppInstanceInfoRequest	Mandatory	MEO → MEPM
QueryAppInstanceInfoResponse	Mandatory	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	Qualifier	Cardinality	Data type	Description
filter	M	1	Filter	Filter to select the application instance(s) about which information is queried. See note.
attributeSelector	M	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: This data type is not specified in the present document.

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	Qualifier	Cardinality	Data type	Description
appInstanceInfo	M	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 notes 1 and 2.

NOTE 1: The lower cardinality is 0 since there may be no matches to the provided filter.
NOTE 2: This data type is not specified in the present document.

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 MEPM and the MEO.

Table 6.3.1.6.1-1: Query application lifecycle operation status operation

Message	Requirement	Direction
QueryAppLcmOperationStatusRequest	Mandatory	MEO → MEPM
QueryAppLcmOperationStatusResponse	Mandatory	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	Qualifier	Cardinality	Data type	Description
lifecycleOperationOccurrenceId	M	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	Qualifier	Cardinality	Data type	Description
operationStatus	M	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 mobile edge application instance.

A mobile edge application instance may be terminated gracefully or forcefully. Graceful termination means that the Mobile Edge Platform Manager gives time to the mobile edge application for application level termination, and after the mobile edge application has terminated in application level, the mobile edge system releases the resources used by the mobile edge application. Forceful termination means that the Mobile Edge Platform Manager immediately shuts down the mobile edge application and releases the resources. A time interval is specified for graceful termination, after the timer specified by the time interval expires, the Mobile Edge Platform Manager will shut down the mobile edge application and release the resources. The graceful termination requires that the mobile edge 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	Qualifier	Cardinality	Data type	Description
appInstancelId	M	1..N	String	Identifier of the mobile edge application instance to be terminated.
terminationType	M	1..N	Enum (inlined)	<p>Signals whether FORCEFUL or GRACEFUL termination is requested.</p> <p>In case of FORCEFUL termination, the mobile edge application is shut down immediately, and resources are released. Note that if the mobile edge application is still in service, this may adversely impact user experience.</p> <p>In case of GRACEFUL termination, the mobile edge system gives time to the mobile edge application for application level termination (e.g. via Mp1 interaction). Once this was successful, or after a timeout, the mobile edge system shuts down the mobile edge application and releases the resources.</p> <p>If the mobile edge application does not support Mp1, the terminationType shall be set to FORCEFUL termination.</p>
gracefulTerminationTimeout	O	0..N	Int	<p>The time interval given to mobile edge application for application level termination during graceful termination, before shutting down the mobile edge application and releasing the resources.</p> <p>If not given, it is expected that the mobile edge system waits for the successful application level termination, no matter how long it takes, before shutting down the mobile edge application and releasing the resources (see note).</p> <p>Minimum timeout or timeout range are specified by the application vendor defined in the AppD.</p> <p>Not relevant in case of forceful termination.</p>
NOTE: This implies that no mobile edge application shutdown and resource release will be attempted if taking the mobile edge 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	Qualifier	Cardinality	Data type	Description
lifecycleOperationOccurrenceId	M	1	String	The identifier of the mobile edge application lifecycle operation occurrence.

6.3.1.7.4 Operation results

In case of success, the mobile edge application instance has been terminated, resources used by the mobile edge 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	Qualifier	Cardinality	Data type	Description
appInstanceid	M	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.

6.3.2 Application lifecycle change notification interface

6.3.2.1 Description

This interface allows the Mobile Edge Orchestrator to subscribe to notifications sent by the Mobile Edge Platform Manager which are related to mobile edge application lifecycle change, as well as the creation/deletion of an application instance identifier and the associated instance of a AppInstanceInfo. It further allows the Mobile Edge Platform Manager to provide such notifications to the subscriber.

6.3.2.2 Subscribe operation

6.3.2.2.1 Description

This operation enables the MEO to subscribe with a filter for the notifications sent by the MEPM which are related to mobile edge application lifecycle changes, as well as creation/deletion of mobile edge application instance identifiers and the associated AppInstanceInfo instances.

Table 6.3.2.2.1-1 lists the information flow exchanged between the MEPM and the MEO.

Table 6.3.2.2.1-1: Subscribe operation

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

6.3.2.2.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.2.2.2-1.

Table 6.3.2.2.2-1: Subscribe operation input parameters

Attribute name	Qualifier	Cardinality	Data type	Description
filter	M	1	Filter	Input filter for selecting e.g. the application instances of interest and the specific types of changes. See notes 1 and 2.
NOTE 1: 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.				
NOTE 2: Refer to clause 6.3.1.5.2. This data type is not specified in the present document.				

6.3.2.2.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.2.2.3-1.

Table 6.3.2.2.3-1: Subscribe operation output parameters

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

6.3.2.2.4 Operation results

After successful subscription, the consumer (MEO) is registered to receive notifications related to application lifecycle changes, as well as creation/deletion of application instance identifiers and the associated AppInstanceInfo instances.

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 consumer.

6.3.2.3 Notify operation

6.3.2.3.1 Description

This operation notifies a subscriber about events related to application lifecycle changes, as well as creation/deletion of application instance identifiers and the associated AppInstanceInfo instances.

This operation distributes notifications to subscribers. It is a one-way operation issued by the producer (MEPM) that may not be invoked as an operation by the consumer (MEO). In order to receive notifications, the consumer (MEO) has to perform an explicit Subscribe operation beforehand.

Table 6.3.2.3.1-1 lists the information flow exchanged between the MEPM and the MEO.

Table 6.3.2.3.1-1: Notify operation

Message	Requirement	Direction
Notify	Mandatory	MEPM → MEO

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

- AppLifecycleChangeNotification.
- AppInstanceIdentifierCreationNotification.
- AppInstanceIdentifierDeletionNotification.

6.3.3 Application package management interface

6.3.3.1 Fetch on-boarded application package operation

6.3.3.1.1 Definition

This operation enables the MEPM to fetch on-boarded 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: FetchAppPackage operation

Message	Requirement	Direction
FetchAppPackageRequest	Mandatory	MEPM → MEO
FetchAppPackageResponse	Mandatory	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: FetchAppPackage operation input parameters

Attribute name	Qualifier	Cardinality	Data type	Description
onboardedAppPkgId	M	1	String	Identifier of the on-boarded 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: FetchAppPackage operation output parameters

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

6.3.3.1.4 Operation results

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

6.3.3.2 Query application package 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.

Table 6.3.3.2.1-1: Query application package operation

Message	Requirement	Direction
QueryAppPackageRequest	Mandatory	MEPM → MEO
QueryAppPackageResponse	Mandatory	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	Qualifier	Cardinality	Data type	Description
queryFilter	M	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	M	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: This data type is not specified in the present document.				

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	Qualifier	Cardinality	Data type	Description
queryResult	M	0..N	AppPackageInfo	Details of the on-boarded application packages matching the input filter. If attributeSelector is present, only the attributes listed in attributeSelector will be returned for the selected entities. See note.
NOTE: This data type is not specified in the present document.				

6.3.3.2.4 Operation results

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 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 MEO and the MEPM.

Table 6.3.3.3.1-1: Subscribe operation

Message	Requirement	Direction
SubscribeRequest	Mandatory	MEPM → MEO
SubscribeResponse	Mandatory	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	Qualifier	Cardinality	Data type	Description
inputFilter	M	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. See note.
NOTE: Refer to clause 6.3.3.2.2. This data type is not specified in the present document.				

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	Qualifier	Cardinality	Data type	Description
subscriptionId	M	1	String	Identifier of the subscription realized.

6.3.3.3.4 Operation results

After successful subscription, the 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 MEPM.

6.3.3.4 Notify Application Package operation

6.3.3.4.1 Definition

This operation distributes notifications to subscribers. It is a one-way operation issued by the MEO that may not be invoked as an operation by the consumer (MEPM).

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

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

- AppPackageOnBoardingNotification.
- AppPackageStateChangeNotification.

Table 6.3.3.4.1-1: Notify operation

Message	Requirement	Direction
Notify	Mandatory	MEO → MEPM

6.3.3.5 Onboarding operation

6.3.3.5.1 Definition

This operation will on-board 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	Qualifier	Cardinality	Data type	Description
name	M	1	String	Name of the application package to be on-boarded.
version	M	1	String	Version of the application package to be on-boarded.
provider	M	1	String	Provider of the application package to be on-boarded.
checksum	M	1	Not specified.	Checksum of the on-boarded application package.
userDefinedData	O	0..N	KeyValuePair	User defined data for the application package. See note 1.
appPackagePath	M	1	URL	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 URL 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	Qualifier	Cardinality	Data type	Description
onboardedAppPkgId	M	1	String	Identifier of the on-boarded the Application package.
appDId	M	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 onboardedAppPkgId 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	Qualifier	Cardinality	Data type	Description
onboardedAppPkgId	M	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 or is in deletion pending state, 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	Qualifier	Cardinality	Data type	Description
onboardedAppPkgId	M	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 or is in deletion pending state, this operation will return an error.

6.3.3.8 Query operation

6.3.3.8.1 Definition

This operation will enable the OSS to query the MEO concerning details of one or more application package(s).

Table 6.3.3.8.1-1 lists the information flow exchanged between the OSS and the MEO.

Table 6.3.3.8.1-1: Query operation

Message	Requirement	Direction
QueryAppPkgRequest	Mandatory	OSS → MEO
QueryAppPkgResponse	Mandatory	MEO → OSS

6.3.3.8.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.8.2-1. The mechanism of filter is not specified in the present document.

Table 6.3.3.8.2-1: QueryAppPackage operation input parameters

Attribute name	Qualifier	Cardinality	Data type	Description
filter	M	1	Filter	Filter defining the application packages on which the query applies, based on attributes of the application packages. It may also be used to specify one or more application packages to be queried by providing their identifiers.
attributeSelector	M	0..N	String	Provides a list of attribute names of the application package. If present, only these attributes will be returned for application package(s) matching the filter. If absent, the complete application package will be returned.
NOTE: Refer to clause 6.3.3.2.2. This data type is not specified in the present document.				

6.3.3.8.3 Output parameters

The output parameters returned by the operation shall follow the indications provided in table 6.3.3.8.3-1.

Table 6.3.3.8.3-1: Query operation output parameters

Attribute name	Qualifier	Cardinality	Data type	Description
queryResult	M	0..N	AppPackage Info	Details of the on-boarded Application package(s) matching the input filter. See note.
NOTE: This data type is not specified in the present document.				

6.3.3.8.4 Operation results

After success 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 packages that the consumer has access to and that matches the filter will be returned.

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 there is no instantiated application instance using it.

An application package in the deletion pending state is no longer enabled or disabled. It is not allowed to instantiate application instance(s) using an application in the deletion pending state.

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	Qualifier	Cardinality	Data type	Description
onboardedAppPkgId	M	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.

If one of the applications is still in use and is marked as delete pending, this will be indicated as part of the operation result and it will be deleted once all associated and instantiated applications are terminated.

6.3.3.10 Abort application package deletion operation

6.3.3.10.1 Description

This operation enables the OSS to abort the deletion of an application package that is in deletion pending state.

Table 6.3.3.10.1-1 lists the information flow exchanged between the OSS and the MEO.

Table 6.3.3.10.1-1 Abort application package deletion operation

Message	Requirement	Direction
AbortPackageDeletionRequest	Mandatory	OSS → MEO
AbortPackageDeletionResponse	Mandatory	MEO → OSS

6.3.3.10.2 Input parameters

The input parameters sent when invoking the operation shall follow the indications provided in table 6.3.3.10.2-1.

Table 6.3.3.10.2-1: Abort VNF Package deletion operation input parameters

Attribute name	Qualifier	Cardinality	Data type	Description
onboardedAppPkgId	M	1	String	Identifier of the onboarded application package of which the deletion is requested to be aborted.

6.3.3.10.3 Output parameters

None.

6.3.3.10.4 Operation results

After successful operation, an onboarded application package that has been in deletion pending state is in disabled state.

It is an error to invoke the operation for an onboarded application package that is not in deletion pending state.

Annex A (normative): 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

Once an application package is onboarded, it has two elementary states:

- operational state (enabled/disabled); and
- usage state (In use/Not in use).

The Deletion Pending state is used when an application package is deleted, but it is in use state.

The transition from one state to the other is indicated in figure A.1 with the given operation name.

The query application package operation is available in all states.

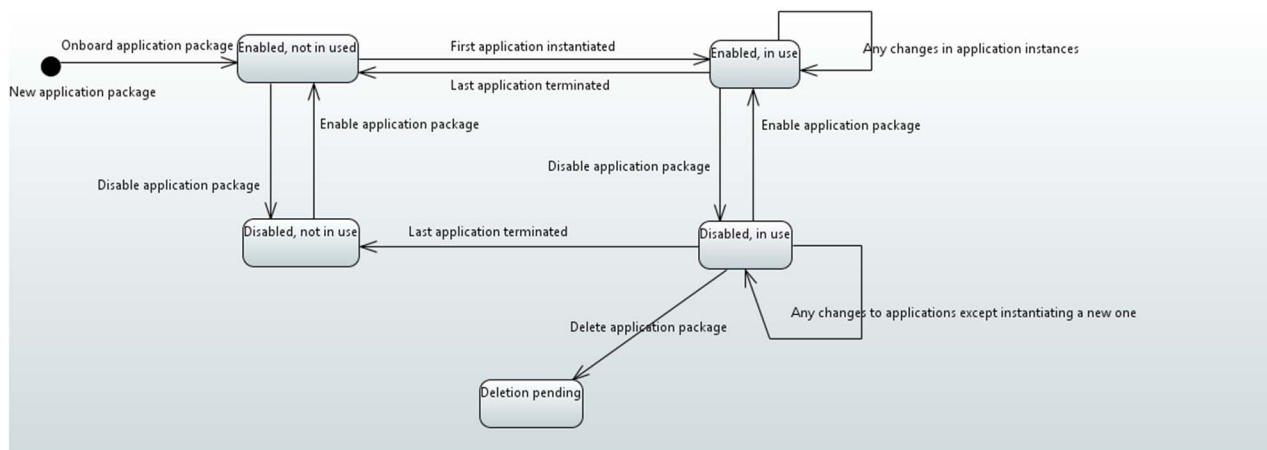


Figure A.1: Application package state model

Annex B (informative): Bibliography

- ETSI GS MEC 003: "Mobile Edge Computing (MEC); Framework and Reference Architecture".
- ETSI GS NFV-IFA 013 (V2.1.1): "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