Draft ETSI GS MEC 040 V3.0.10 (2022-07)

Multi-access Edge Computing (MEC); Federation enablement APIs

Disclaimer: This DRAFT is a working document of ETSI ISG MEC. It is provided for information only and is still under development within ETSI ISG MEC. ETSI and its Members accept no liability for any further use/implementation of this Specification.

Non-published MEC drafts stored in the "Open Area" are working documents, these may be updated, replaced, or removed at any time

Do not use as reference material.

Disclaimer

Do not cite this document other than as "work in progress".

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

Approved and published Specifications and reports for implementation of the MEC system shall be obtained via the ETSI Standards search page at: http://www.etsi.org/standards-search

Reference
DGS/MEC-0040FederationAPI

Keywords
API, Interworking, MEC, service

ETSI

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

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

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights 5
Foreword 5
Modal verbs terminology 5
Executive summary 5
Introduction 5
1 Scope 6
2 References 6
2.1 Normative references 6
2.2 Informative references 6
3 Definition of terms, symbols and abbreviations 7
3.1 Terms 7
3.2 Symbols 7
3.3 Abbreviations 7
4 Overview 7
4.1 Introduction 7
4.2 GSMA Operator Platform and its interfaces 8
5 Description of the services (informative) 8
5.1 Federation enablement service introduction 8
5.2 Sequence diagrams 9
5.2.1 Introduction 9
5.2.2 Registration/Update/Deregistration of MEC system(s) to the federation 9
5.2.2.1 Registration 9
5.2.2.2 Update 10
5.2.2.3 Deregistration 10
5.2.3 MEC system discovery 11
5.2.4 MEC application instance discovery 12
5.2.5 MEC service discovery 13
5.2.6 Application package management and Application instance lifecycle management 14
5.2.6.1 Introduction 14
5.2.6.2 Onboarding application package 14
5.2.6.3 Application Instantiation 15
5.2.7 Providing MEC system-wide MEC application instance information updates to MEF 16
5.2.8 Forwarding MEC system-wide MEC application instance information updates to another MEC system of MEC federation 17
5.2.X1 REST based subscribe-notify model 17
5.2.X1.x1 Subscribing to federation event notifications 17
5.2.X1.x2 Receiving notification on expiry of federation event subscription 18
5.2.X1.x3 Updating subscription for federation event notifications 19
5.2.X1.x4 Unsubscribing from federation event notifications 19
5.2.X2 Receiving MEC system registration/update notifications 20
5.2.X3 Receiving MEC application instance registration/ registration update/ deregistration notifications 20
6 Data model 21
6.1 Introduction 21
6.2 Resource data types 21
6.2.1 Introduction 21
6.2.2 Type: SystemInfo 21
6.3 Subscription data types 21
6.3.1 Introduction 21
6.3.2 Type: SystemUpdateNotificationSubscription 21
6.4 Notifications data types 22
6.4.1 Introduction 22
6.4.2 Type: SystemUpdateNotification 22

6.5 Referenced structured data types 22
6.5.1 Introduction 22
6.5.2 Type: TimeStamp 23
6.6 Referenced simple data types and enumerations 23
6.6.1 Introduction 23
6.6.YY Type: <TypeNameYY> 23

7 API definition 23
7.1 Introduction 23
7.2 Global definitions and resource structure 23
7.w Resource: <ResourceNameW> 23
7.w.1 Description 23
7.w.2 Resource definition 23
7.w.3 Resource methods 23
7.w.3.1 GET 23
7.w.3.2 PUT 23
7.w.3.3 PATCH 23
7.w.3.4 POST 23
7.w.3.5 DELETE 23

Annex A (informative): Enabling MEC App providers to access MEC federation services 24
A.1 Introduction 24
A.2 OP-NBI & OP-E/WBI API mapping to ETSI MEC specifications 25

Annex B (informative): Complementary material for API utilisation 27

Annex (informative): Bibliography 28

Annex (informative): Change History 29

History 30

Intellectual Property Rights

Essential patents

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

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

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

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

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

Introduction

1 Scope

The present document focuses on the functionalities enabled over the relevant reference points (i.e. Mfm-fed, Mff-fed, and Mfb-fed) to support MEC federation. It describes the information flows, required information, and specifies the necessary operations, data models and API definitions. The present document carefully considers the relevant work of other industry bodies relating to MEC federation (e.g. GSMA OPG, 5GAA, etc.) and all relevant work done in ETSI.

2 References

2.1 Normative references

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

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

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

The following referenced documents are necessary for the application of the present document.

[1] ETSI GS MEC 001: "Multi-access Edge Computing (MEC); Terminology".

2.2 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] GSMA Permanent Reference Document: "Operator Platform Telco Edge Requirements", v1.0, Jun. 2021.

NOTE: Available at https://www.gsma.com/futurenetworks/wp-content/uploads/2021/06/OPG-Telco-Edge-Requirements-2021.pdf.

[i.2] ETSI GR MEC 035: "Multi-access Edge Computing (MEC); Study on Inter-MEC systems and MEC-Cloud systems coordination".

[i.3] ETSI GS MEC 002: "Multi-access Edge Computing (MEC); Phase 2: Use Cases and Requirements".

[i.4] ETSI GS MEC 003: "Multi-access Edge Computing (MEC); Framework and Reference Architecture".

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

[i.6] ETSI GS MEC 010-2: " Multi-access Edge Computing (MEC); MEC Management; Part 2: Application lifecycle, rules and requirements management ".

[i.7] OpenAPI™ Specification.

NOTE: Available at https://github.com/OAI/OpenAPI-Specification.

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

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

App	Application
CHF	Charging Functions
CR	Cloud Resources
E/WBI	East/West Bound Interface
GSMA	GSM Association
LADN	Local Area Data Network
LBO	Local Breakout
MEF	MEC Federator
MNO	Mobile Network Operator
NBI	NorthBound Interface
NR	Network Resources
OEM	Original Equipment Manufacturer
OP	Operator Platform
QoS	Quality-of-Service
SBI	SouthBound Interface
UNI	User Network Interface

4 Overview

4.1 Introduction

The present document specifies Federation enablement APIs that enable the shared usage of MEC services and applications across different systems (e.g. MEC system, Cloud system).

Clause 4 introduces the relevant work of other industry bodies e.g. GSMA OPG.

Clause 5 presents the reference scenarios for the MEC federation, and introduces the functionalities enabled via the relevant reference points (i.e. Mfm-fed, Mff-fed and Mfb-fed). It provides the high-level information flows and describes the necessary operations.

Clause 6 describes the data models that can be exchanged over the Federation enablement APIs, which provide detailed descriptions of all information elements used for MEC federation.

Clause 7 defines the actual Federation enablement APIs providing detailed information of how information elements are mapped into a RESTful API design.

4.2 GSMA Operator Platform and its interfaces

According to the GSMA Permanent Reference Document (PRD), "Operator Platform Telco Edge Requirements" [i.1], an Operator Platform (OP) is a facilitator of subscribers' seamless access to edge applications instantiated within a federation of edge networks involving multiple owners. Such seamless access is needed either when subscribers roam to visited networks or when a partner network is a better choice for edge application instantiation.

The objective of the OP concept is to guide the industry ecosystem, i.e. MNOs, vendors, OEMs and service providers towards shaping a common solution for the exposure of network capabilities. As an initial step, [i.1] provides both an end-to-end definition and requirements of the OP for the support of edge computing. In further details, the GSMA defines OP requirements as well as OP architecture and functional modules. Therefore, aim of GSMA is to engage with standardization and open source communities that will undertake the standard definition of the OP. As depicted in Figure 4.2-1, the following OP interfaces have been defined [i.1]:

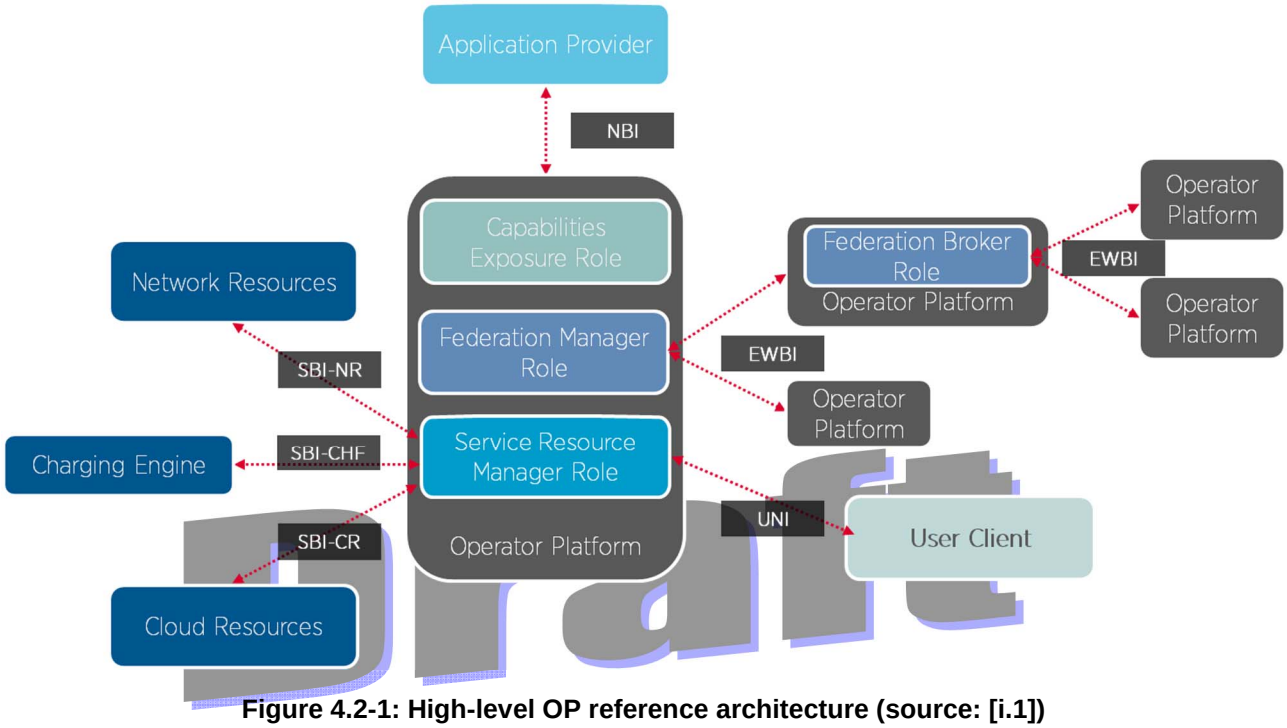


Figure 4.2-1: High-level OP reference architecture (source: [i.1])

- Northbound Interface (NBI);
- Southbound Interface (SBI); Cloud Resources (SBI-CR);
- Southbound Interface (SBI); Network Resources (SBI-NR);
- Southbound Interface (SBI); Charging Functions (SBI-CHF);
- User Network Interface (UNI);
- East/ West Bound Interface (E/WBI).

Editor's note: for the above figure we need a copyright permission from GSMA OPG.

5 Description of the services (informative)

5.1 Federation enablement service introduction

Federation enablement APIs offers services such as discovery, information exchange and application life cycle management to enable the inter-work of one MEC system with another MEC system. The related requirements were carefully studied and extracted from various use cases in ETSI GR MEC 035 [i.2] including V2X services scenario,

multi-operator environment, Application instance transfer between a MEC system and a MEC/Cloud system, connecting different services, immersive AR game scenario, edge service delivery through visited network and edge node sharing.

The extracted requirements are listed as follows, summarized from ETSI GS MEC 002 [i.3].

- MEC system discovery ([Federation-02])
- MEC platform discovery ([Federation-03])
- Information exchange between MEC systems ([Federation-04])
- Information exchange between MEC platforms ([Federation-05])
- Support handling direct/indirect MEC system communication ([Federation-06])
- MEC application discovery ([Federation-07])
- MEC application on-boarding/instantiation ([Federation-08])
- Information exchange among MEC applications ([Federation-09])

NOTE: Reusing the data models and APIs for MEC-Cloud coordination is considered if applicable, but its information flow is out of scope of the present document.

5.2 Sequence diagrams

5.2.1 Introduction

The rest of clause 5.2 introduces the following sequence diagrams based on the extracted requirements.

- Registration of MEC system(s) to the federation (clause 5.2.2)
- Discovery:
  - MEC system discovery (clause 5.2.3)
  - MEC application discovery (clause 5.2.4)
  - MEC service discovery (clause 5.2.5)
- Application package management and Application instance lifecycle management (clause 5.2.6)

NOTE 1: Support handling direct/indirect MEC system communication is satisfied by MEC Federator as defined in ETSI GS MEC 003 [i.4].

NOTE 2: The requirement for registration is based on the premise that multiple MEOs can register to a single MEF.

5.2.2 Registration/Update/Deregistration of MEC system(s) to the federation

5.2.2.1 Registration

Information flow of registration is used for enabling MEO to register its own information to MEC Federator over Mfm reference point. The registered information may contain e.g. MEC system information, etc. The information flow is depicted in Figure 5.2.2.1-1.

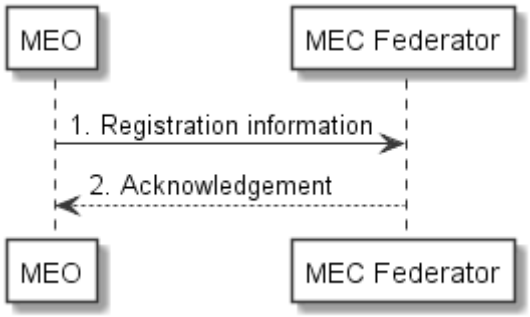


Figure 5.2.2.1-1: Information flow of Registration

Registration procedure consists of the following steps:

1) The MEO sends a registration request to MEC Federator.

2) MEC Federator returns an acknowledgement to MEO.

5.2.2.2 Update

Information flow of update of MEC system(s) to the federation is depicted in Figure 5.2.2.2-1.

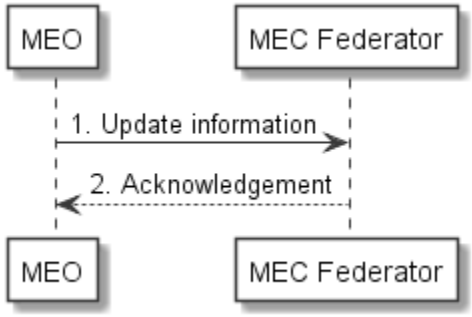


Figure 5.2.2.2-1: Information flow of Update

Update procedure consists of the following steps:

3) The MEO sends an update request to MEC Federator.

4) MEC Federator returns an acknowledgement to MEO.

5.2.2.3 Deregistration

Information flow of deregistration of MEC system(s) from the federation is depicted in Figure 5.2.2.3-1.

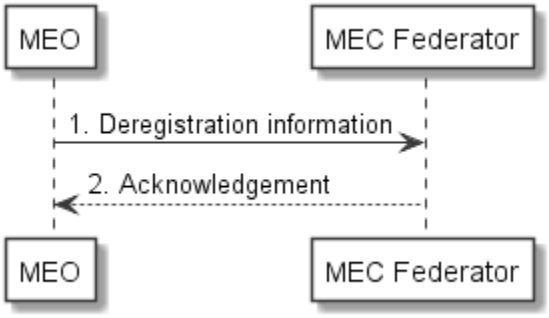


Figure 5.2.2.3-1: Information flow of Deregistration

Deregistration procedure consists of the following steps:

1) The MEO sends an update request to MEC Federator.

2) MEC Federator returns an acknowledgement to MEO.

5.2.3 MEC system discovery

Information flow of MEC system discovery is used for enabling MEO to be aware of another MEC system. MEC system discovery is the primitive and essential procedure for enabling the other functionalities relating to Feature MECFederation. The information flow is depicted in Figure 5.2.3-1.

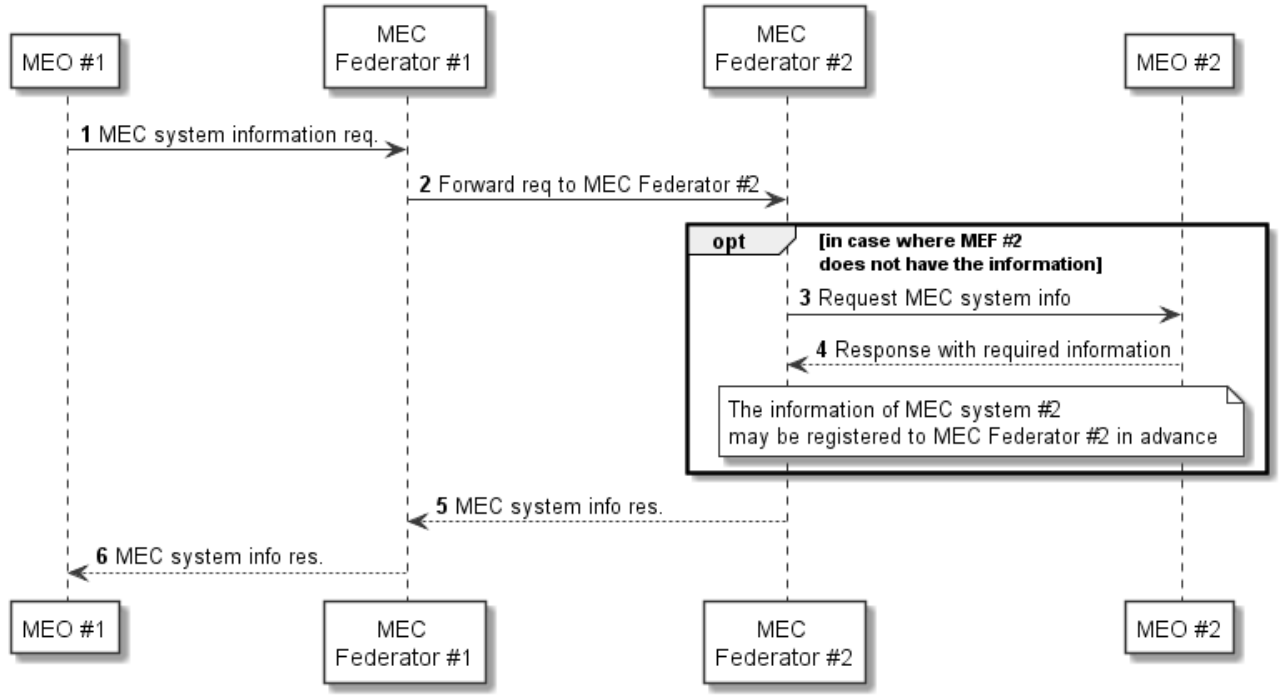


Figure 5.2.3-1: Information flow of MEC system discovery

As a prerequisite of this flow, MEC Federator Discovery is conducted among MEC Federators, which means MEC Federators are aware of each other in advance.

1) The MEO #1 sends a MEC system information request to MEC Federator #1 over Mfm reference point. This request is triggered by MEC platform or MEC Application instance.

2) MEC Federator #1 forwards the request to MEC Federator #2.

3) In case where MEC Federator #2 does not have the desired information (which means MEO #2 does not register its own information in advance), MEC Federator #2 sends a MEC system information request to MEO #2 over Mfm reference point.

4) MEO #2 responds with the information of its own system to MEC Federator #2.

5) MEC Federator #2 forwards the response to MEC Federator #1.

6) MEC Federator #1 forwards the response to MEO #1.

Editor's note: the addition of pub/sub mechanisms still have to be considered as an additional option for MEC system discovery.

5.2.4 MEC application instance discovery

MEC application instance discovery refers to a process triggered by a MEC application instance, which discovers one or more MEC application instances in the MEC federation of the application from which it was instantiated. For example, the discovery may be based on information of a specific MEC application instance or of the corresponding application descriptor. This process is triggered, for instance, in the cases calling for MEC application instance-to-instance communication (e.g. neighboring vehicles communicating with different MEC application instances may need to cooperate via those MEC application instances. grouped users communicating with different MEC application instances may need to communicate with each other via those MEC application instances, or grouped users may be gathered from different MEC systems and served by a single MEC application instance). The information flow is depicted in Figure 5.2.4-1.

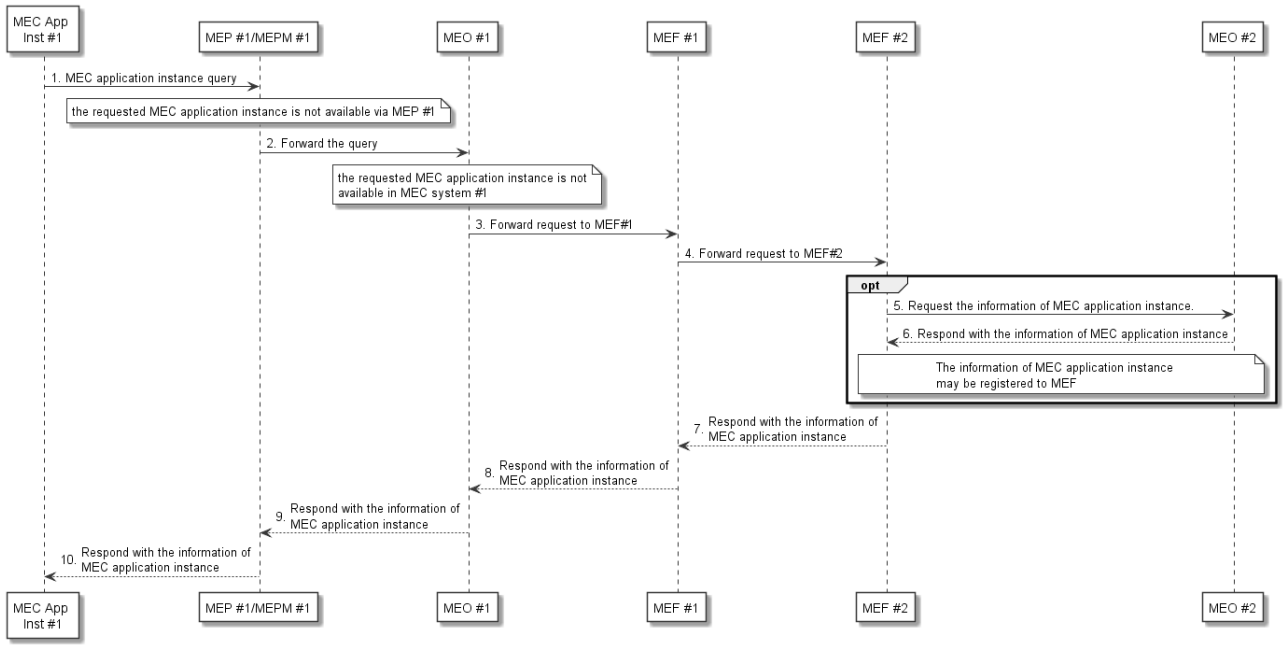


Figure 5.2.4-1: Information flow of MEC application discovery

Procedure of MEC application instance discovery consists of the following steps:

1) MEC application instance #1 sends a query to MEP #1 to discover a MEC application instance.

NOTE 1: MEC application instance #1 may know either the identifier of the requested application instance or identifier of application descriptor.

2) In the case where the desired MEC application instance is not available via MEP #1, MEP #1 forwards the query to MEO #1 via MEPM #1.

NOTE 2: How to handle the query between MEP #1 and MEO #1 is not further specified in the present document.

3) MEO #1 examines if the requested MEC application instance is available in MEC system #1. In case where the MEC application instance is not available in MEC system #1, MEO #1 forwards the query to MEF #1. Otherwise, go to step 9.

4) MEF #1 forwards the query to MEF #2.

5) If the information of MEC application instance, i.e. the list of active MEC application instances, is not registered to MEF #2, MEF #2 forwards the query to MEO #2.

6) MEO #2 responds with the information of MEC application instance(s). If no available MEC application is discovered, void would return.

NOTE 3: In the case where multiple MEC application instances are found, the information of a list of MEC application instances is returned. In this case, the information of the MEC platform associated with each instance might be useful for the selection of MEC application instances. However, for security reasons, the information of MEC platform should be hidden between federated MEC systems. Based on the agreement among federated MEC systems, the information of MEC platform, e.g. available MEC services, can be included in the response.

7) MEF #2 responds with the information of MEC application instance(s) to MEF #1.

8) MEF #1 responds with the information of MEC application instance(s) to MEO #1.

9) MEO #1 responds with the information of MEC application instance(s) to MEP #1 via MEPM #1.

10) MEP #1 responds with the information of MEC application instance(s) to MEC application instance #1.

5.2.5 MEC service discovery

MEC service discovery in a MEC federation can be performed when a MEC system of the MEC federation wants to obtain MEC service availability. This process could be triggered in the case where the service consumer (e.g. a MEC application or a MEC platform of a MEC system the service discovery query originates from) needs the specific MEC service that is not available at the collocated MEC platform. The information flow is depicted in Fig. 5.2.5-1.

NOTE: the desired MEC service could be provided by multiple MEC platforms. The service consumer that triggers the discovery may select one service instance.

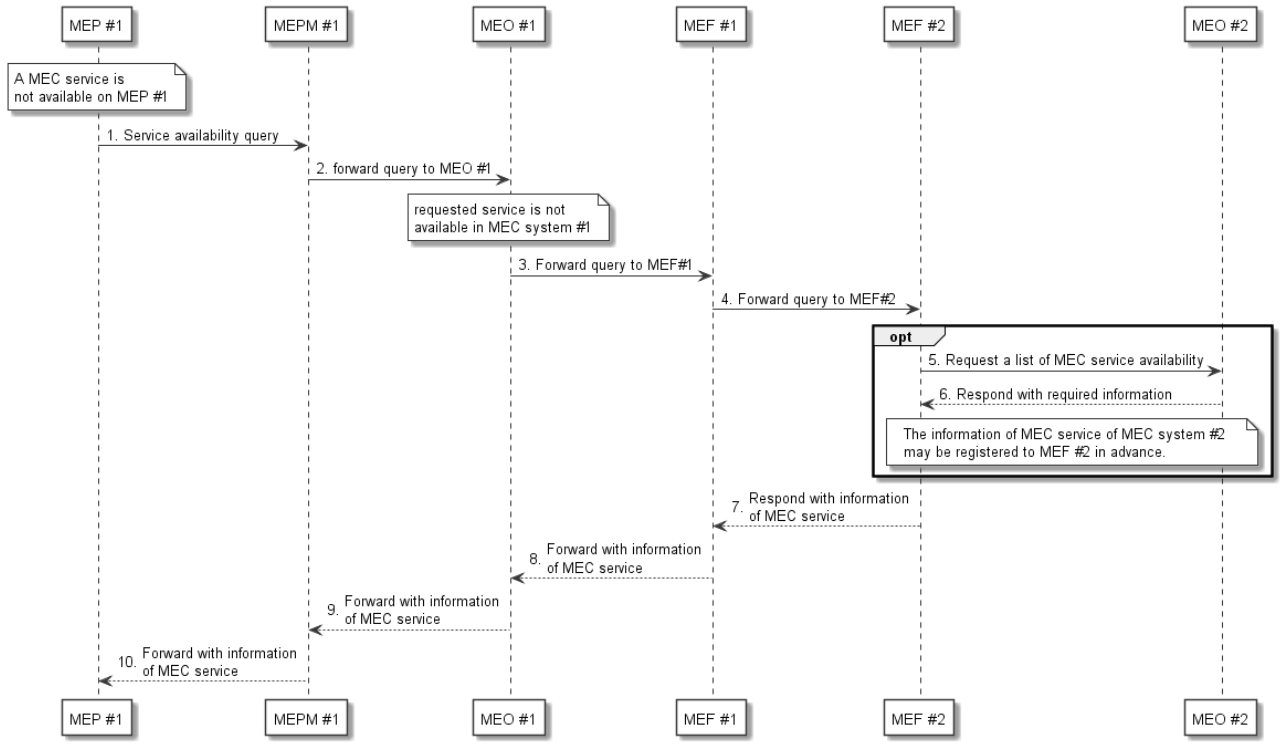


Figure 5.2.5-1: Information flow of MEC service discovery.

Procedure of MEC service discovery consists of the following steps:

1) MEP#1 sends a request to query the availability of a MEC service or a list of MEC services.
2) MEPM #1 forwards the query to MEO #1.

NOTE: How to handle the request between MEP, MEPM and MEO is not further specified in the present document.

3) MEO #1 examines if the requested service(s) are available in MEC system #1. In case where the service(s) are not available, MEO #1 forwards the query to MEF #1. Otherwise, MEO#1 responds with the information of MEC service(s) to MEPM #1.

NOTE: The case where the requested service(s) are available in the MEC system is out of scope of the present document.

4) Subject to federation agreements and operator policies, MEF #1 forwards the query to MEF #2.
5) Optionally, MEF #2 forwards the query to MEO #2. Otherwise, go to step 7.

NOTE: MEF is assumed to subscribe MEC service availability.

Editor's note: It is FFS with regard to the case where MEF is not able to subscribe MEC service availability.

6) MEO #2 responds with the information of MEC service(s). If the requested MEC service(s) are not available in MEC system #2, MEO #2 returns void.

NOTE: For security reasons, the information of the corresponding MEC platform information should be hidden between the federated MEC systems.

7) MEF #2 responds with the information of MEC service(s) to MEF #1.
8) MEF #1 forwards the information of MEC service(s) to MEO #1.
9) MEO #1 forwards the information of MEC service(s) to MEPM #1.
10) MEPM #1 forwards the information of MEC service(s) to MEP #1.

Editor's note: How to realize MEC service discovery using a pub/sub mechanism is FFS.

5.2.6 Application package management and Application instance lifecycle management

5.2.6.1 Introduction

The overall procedures for Application package management and Application instance lifecycle management follow clause 5.2 "Application management" and clause 5.3 "Application instance lifecycle management" in ETSI GS MEC 010-2 [i.6]. The difference is that all requests are forwarded to the external MEC system in the MECFederation through MEC federator with the information obtained in clauses "5.2.2. " and "5.2.3" . This clause describes onboarding an application package and application instantiation, and it can be assumed that the rest of other procedures are also supported through Mfm, Mff interface, and MEC federator, similar to those introduced here. These requests can be triggered at an entity (e.g. OSS) that is connected to application provider.

Editor's note: To merge with "MEC host discovery" will be handled.

Application package management:

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

Application instance lifecycle management:

- Application instantiation
- Application termination
- Application operation

5.2.6.2 Onboarding application package

The message flow of on-boarding application package is used to make application package available to the MEC system in the MECFederation. On-boarding request is triggered by Client (e.g. OSS) of MEC system #1 and forwarded to federated MEC system. The detailed description of this flow is depicted in figure 5.2.6.2-1

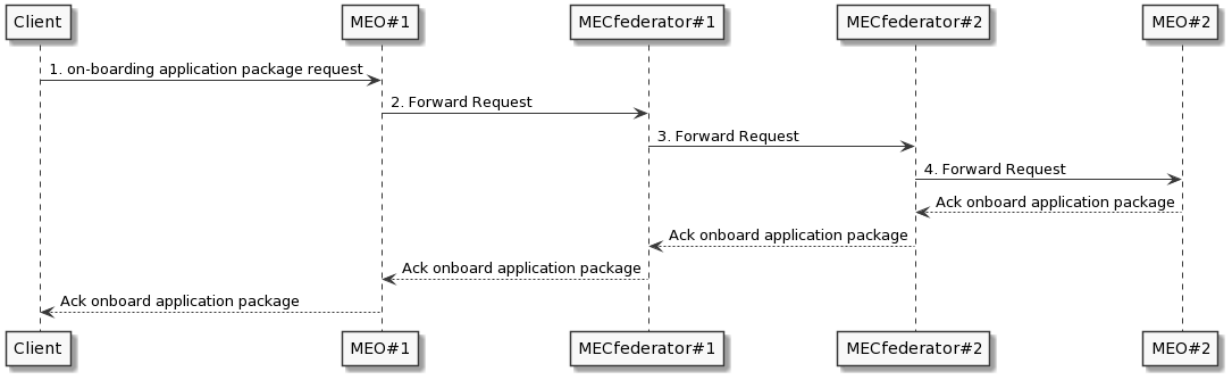


Figure 5.2.6.2-1: On-boarding application package in MEC federation

1. Client sends an on-boarding application package request to the MEO #1 with the information for MECFederation. The information may be based on the result of 5.2.3 MEC system discovery.
2. If information for MECFederation is included, MEO #1 forwards the request to MEC federator #1 and continue with the rest of the steps. Otherwise, follow the procedures of 5.2.2 in ETSI MEC 10-2

   Editor's note: Step 1& 2 will impact on ETSI MEC010-2
3. MEC federator #1 forwards the request to MEC federator#2
4. MEC federator #2 forwards the request MEO #2 based on included MEC system information.
5. MEO #2 performs the actions described in step 1 of 5.2.2 of the ETSI MEC 10-2[i.5] and MEC #2 returns an acknowledgement to MEC federator #2

6-8. The acknowledgement is forward to Client via MEC federator #1 and MEO #1.

5.2.6.3 Application Instantiation

The message flow of application instantiation is used to instantiate an application instance in the MEC system.

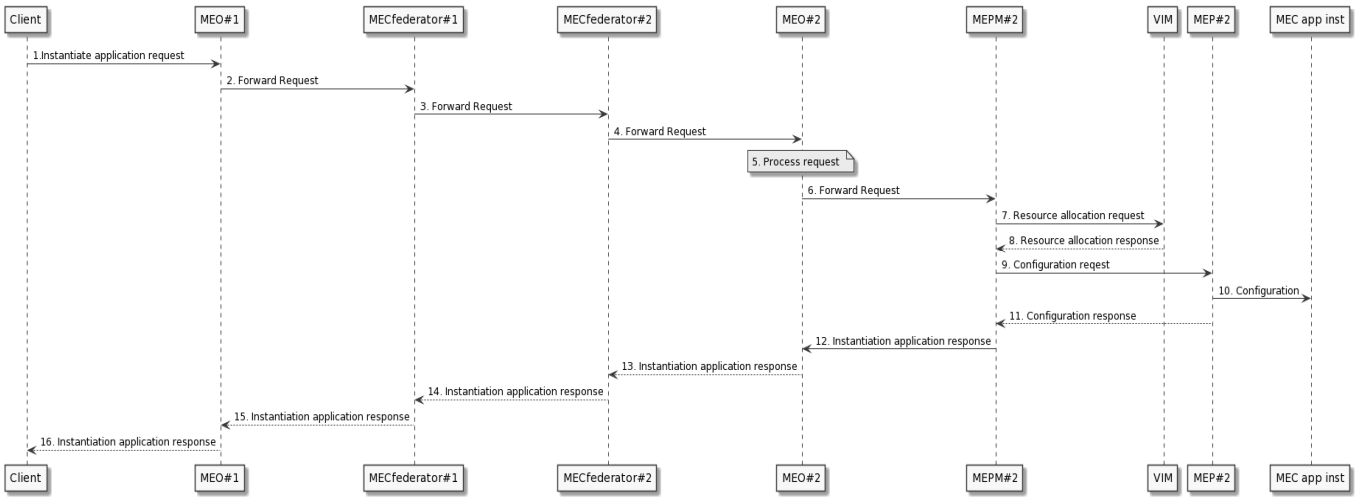


Figure 5.2.6.3-1: Application instantiation flow in MEC federation

1. Client sends an instantiate application request to the MEO #1 with the information for MECFederation. The information may be based on the result of 5.2.3 MEC system discovery.

2. If information for MECFederation is included, MEO #1 forwards the request to MEC federator #1 and continue with the rest of the steps. Otherwise, follow the procedures of 5.3.2 in ETSI MEC 10-2

   Editor's note: Step 1& 2 will impact on ETSI MEC010-2

3. MEC federator #1 forwards the request to MEC federator #2

4. MEC federator#2 forwards the request to MEO #2.

5. MEO #2 checks the application instance configuration data, and authorizes the request and selects the MEC host (and corresponding MEC platform manager). If necessary, MEO #2 performs the actions described in step 3 of clause 5.3.1 of the ETSI MEC 10-2[i.5].

6-12. Steps 6-12 follow the same procedures as described in steps 4-10 of clause 5.3.1 of the ESTI MEC 10-2 [i.5].

13. MEO #2 forwards the response to MEC federator#2 including the results of the instantiation operation and the application instance ID if there is.

14-16. The response is forwarded to Client through MEC federator #2 and MEC federator #1.

5.2.7 Providing MEC system-wide MEC application instance information updates to MEF

Figure 5.2.7-1 explains the procedure of providing MEC application instance information updates to the MEF within the MEC system. This procedure can be triggered by the instantiation of a MEC application or a new or updated MEC application instance registration.

Editor's note: It is FFS how MEO obtains the information of an MEC application instance registration to the MEC platform. The impacts, if any, on MEC 010-2 and MEC 011 may need to be addressed in separate contributions.

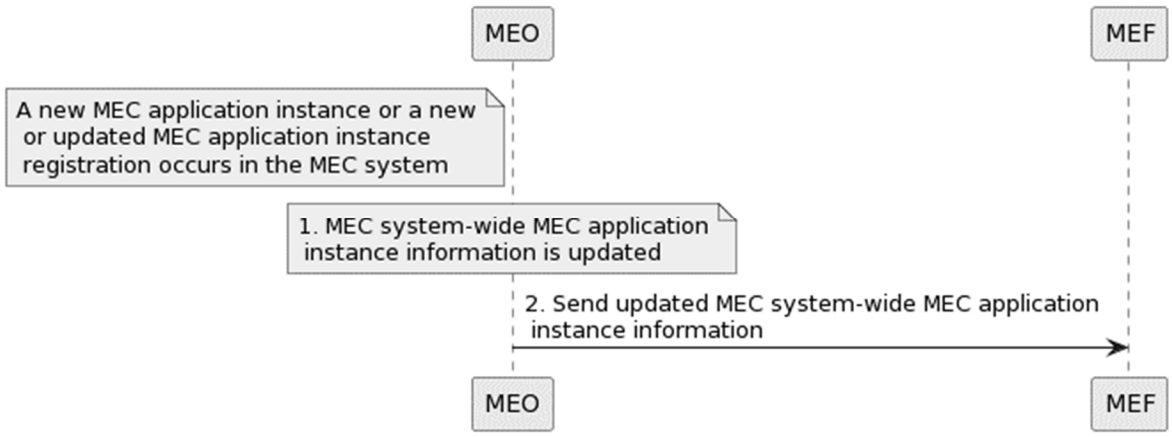


Figure 5.2.7-1: Information flow of providing MEC application instance information updates to MEF.

The procedure of a MEO providing MEC system-wide MEC application instance information updates to MEF when, within the MEC system, there is a MEC application instantiation by the MEC system, or a MEC application instance registers with a MEC platform or updates its registration information, consists of the following steps:

1) MEC system-wide MEC application instance information is updated.
2) The MEO forwards the updated MEC system-wide MEC application instance information to the MEF.

Editor's note: It is FFS what MEC application instance information would be shared in the MEC federation. Characterization of MEC system-wide MEC application instance information as MEC system-private is FFS.

Editor's note: it is FFS whether MEC application instance information masking is done in MEO, MEF or elsewhere.

NOTE: The definition of specific authorization, communication periodicity and filtering policies, that are, instead, related to business agreements among federating entities, is outside the scope of this document.

5.2.8 Forwarding MEC system-wide MEC application instance information updates to another MEC system of MEC federation

Figure 5.2.8-1 illustrates the inter-MEF communication needed within a MEC federation to share the permissible (e.g. per MEC system authorization policy) MEC system-wide MEC application instance information updates.

Figure 5.2.8-1: Information flow of providing permissible MEC system-wide MEC application instance information updates to another MEC system of the MEC federation.

The procedure of forwarding MEC system-wide MEC application instance information updates to another MEC system of MEC federation consists of the following steps:

1) MEF 1 sends the updated MEC system-wide MEC application instance information to MEF 2.

Editor's note: It is FFS what MEC application instance information would be shared in the MEC federation. Characterization of MEC system-wide MEC application instance information as MEC system-private is FFS.

NOTE: The definition of specific authorization, communication periodicity and filtering policies, that are, instead, related to business agreements among federating entities, is outside the scope of this document.

5.2.X1 REST based subscribe-notify model

Editor's note: the order of this clause may change depending on other sequence diagrams that will be approved in future. This clause would be located after other request/response diagrams.

5.2.X1.x1 Subscribing to federation event notifications

To receive a notification of federation event, the Client creates a subscription to the event as depicted in Figure 5.2.X1.x1-1.

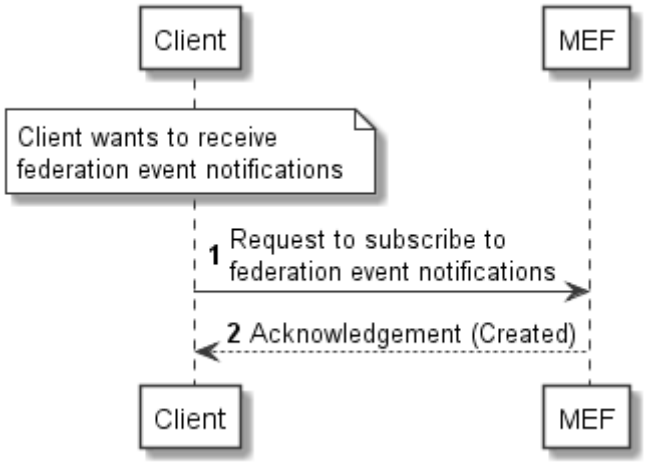


Figure 5.2.X1.x1-1: Information flow of subscribing to federation event notifications

Subscribing to federation event notification consists of the following steps.

When the Client wants to receive notifications about MEC system registration/update, it creates a subscription to federation event:

1) The Client sends a request for subscribing to federation event notifications.

2) The MEF returns an acknowledgement to the Client.

Editor's note: The set of federation events needs to be specified.

5.2.X1.x2 Receiving notification on expiry of federation event subscription

An expiry time for a federation event subscription may be defined. In case expiry time is used, prior to the expiry, a notification is sent to the subscriber MEO. The scenario where the MEC Federator sends the notification is depicted in Figure 5.2.X1.2-1.

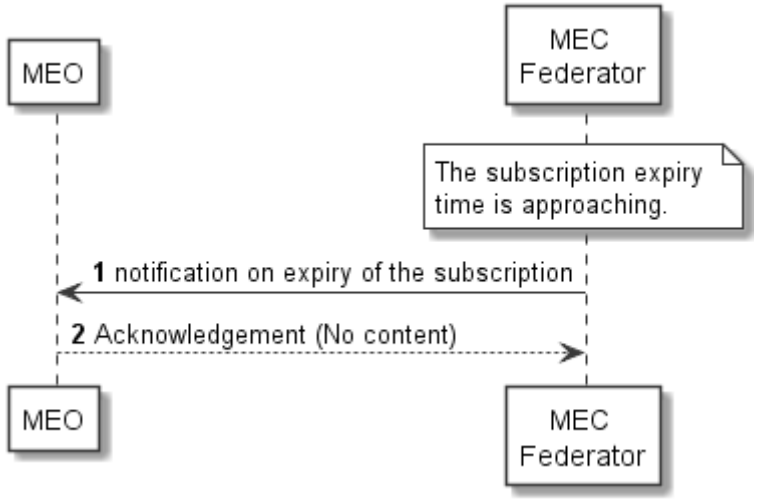


Figure 5.2.X1.x2-1: Information flow of receiving notification on expiry of federation event subscription

Receiving notification on expiry of federation event subscription consists of the following steps:

1) The MEC Federator sends a notification on expiry of the subscription when the subscription expiry time is approaching.

2) The MEO returns an acknowledgement to the MEC Federator.

5.2.X1.x3 Updating subscription for federation event notifications

Figure 5.2.X1.x3-1 presents the scenario where the MEO needs to update an existing subscription for federation event notification.

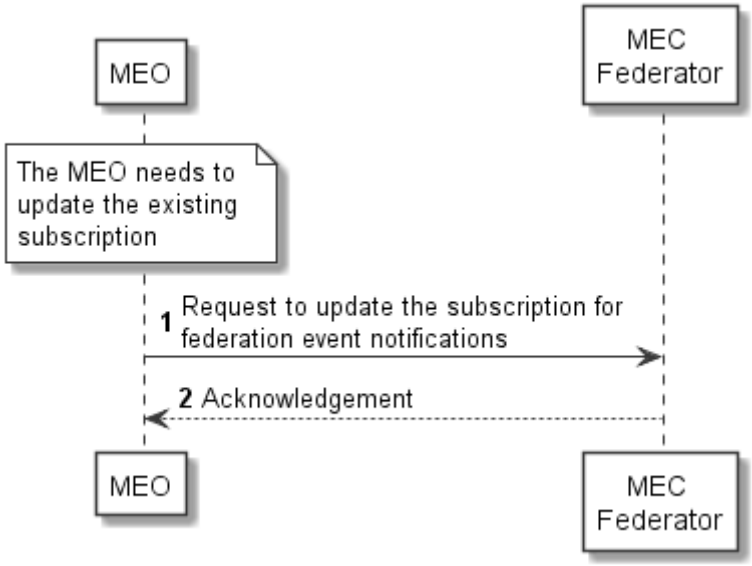


Figure 5.2.X1.x3-1: Information flow of updating subscription for federation event notifications

Updating subscription for federation event notification consists of the following steps:

1) The MEO sends a request for updating subscription for federation event notifications.
2) The MEC Federator returns an acknowledgement to the MEO.

5.2.X1.x4 Unsubscribing from federation event notifications

Figure 5.2.X1.x4-1 shows the scenario of deleting the subscription when the MEO does not want to receive federation event notification after subscribing to the event.

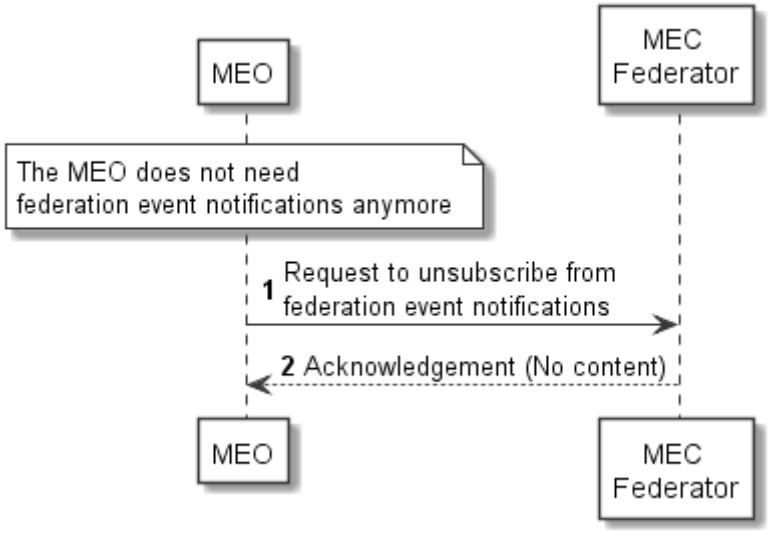


Figure 5.2.X1.x4-1: Information flow of unsubscribing from federation event notifications

Unsubscribing federation event notification consists of the following steps:

1) The MEO sends a request for unsubscribing from federation event notifications.

2) The MEC Federator returns an acknowledgement to the MEO.

5.2.X2 Receiving MEC system registration/update notifications

Figure 5.2.X2-1 presents the scenario of MEC system registration/update notification.

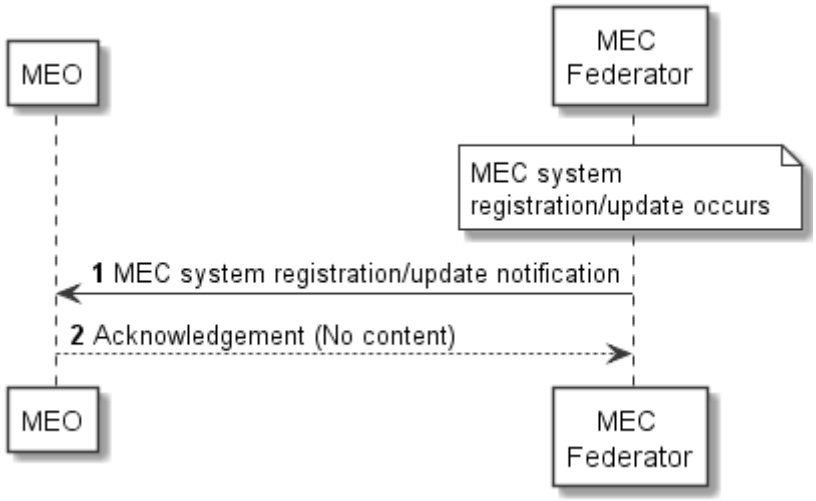


Figure 5.2.X2-1: Information flow of receiving MEC system registration/update notification

Receiving MEC system registration/update notification consists of the following steps:

1) When MEC system registration/update occurs, the MEC Federator sends a notification of registration/update of other MEC system to the subscriber MEO.

2) The MEO returns an acknowledgement to MEC Federator.

5.2.X3 Receiving MEC application instance registration/ registration update/ deregistration notifications

Figure 5.2.X3-1 presents the scenario of MEC application registration/registration update/deregistration notifications.

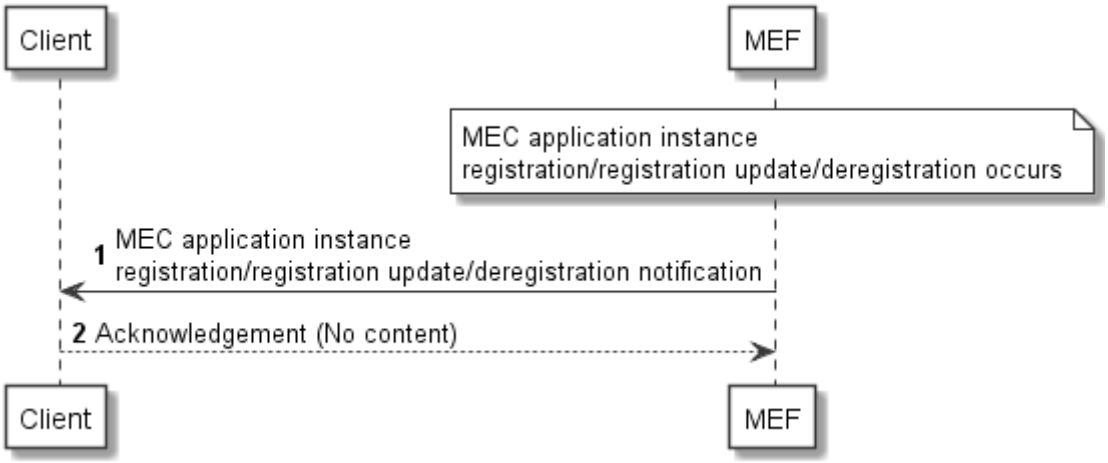


Figure 5.2.X3-1: Information flow of receiving MEC application instance registration/registration update/deregistration notification

Receiving MEC application instance registration/registration update/deregistration notification consists of the following steps:

1) When MEC application instance registration/registration update/deregistration occurs, the MEF sends a notification of registration/registration update/deregistration of other MEC system to the subscriber Client.
2) The Client returns an acknowledgement to MEF.

Editor's note: How the Federator found out that an app instance has been registered/updated/deregistration is FFS.

6 Data model

6.1 Introduction

The following clauses define the data types common to the APIs specified in the present document.

6.2 Resource data types

6.2.1 Introduction

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

6.2.2 Type: SystemInfo

This type represents an information provided by MEC orchestrator as a part of the “Registration of MEC system to the federation” introduced in Clause 5.2.2.1.

The attributes of the SystemInfo shall follow the indications provided in Table 6.2.2-1.

Table 6.2.2-1: Attributes of SystemInfo

Attribute name	Data type	Cardinality	Description
systemId	String	1	Identifier of the MEC system. For the uniqueness of the identifier across the federated MEC systems.
systemName	String	1	The name of the MEC system. This is how the MEC system identifies other MEC systems
systemProvider	String	1	Provider of the MEC system.
endpoint	EndPointInfo	0..1	Endpoint information (e.g. URI, FQDN, IP address) of MEC federator.

Editor’s Note: it is FFS on how to check uniqueness of systemId identifiers and how to handle that according to OPG.

6.3 Subscription data types

6.3.1 Introduction

This clause defines data structures for subscriptions.

6.3.2 Type: SystemUpdateNotificationSubscription

This type represents a subscription to the notifications from the MEC federator related to information update of the MEC systems in the MEC federation.

The attributes of the SystemUpdateNotificationSubscription shall follow the indications provided in table 6.3.2-1.

Table 6.3.2-1: Attributes of SystemUpdateNotificationSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "SystemUpdateNotificationSubscription".
callbackReference	Uri	1	URI selected by the MEC orchestrator to receive notifications on the subscribed MEC system information updates in the MEC federation. This shall be included in both the request and the response.
_links	Structure (inlined)	0..1	Object containing hyperlinks related to the resource. This shall only be included in the HTTP responses.
>self	LinkType	1	Self-referring URI. The URI shall be unique within the MEC Federation Enablement API as it acts as an ID for the subscription (SubscriptionId).
systemId	String	0..N	Identifier(s) to uniquely specify the target MEC system(s) for the subscription. If absent, the subscription should include all MEC systems in the MEC federation.
expiryDeadline	TimeStamp	0..1	The expiration time of the subscription determined by the MEC Federation Enablement Service.

6.4 Notifications data types

6.4.1 Introduction

This clause defines data structures that define notifications.

6.4.2 Type: SystemUpdateNotification

This type represents the information that the MEC federator notifies the subscribed MEC orchestrator about the information update of the MEC systems in the MEC federation.

The attributes of the SystemUpdateNotification shall follow the indications provided in table 6.4.2-1.

Table 6.4.2-1: Attributes of SystemUpdateNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "SystemUpdateNotification".
updatedSystemInfo	SystemInfo	1..N	Updated information of the MEC system(s) in the MEC federation.
_links	Structure (inlined)	1	Object containing hyperlinks related to the resource.
>subscription	LinkType	1	A link to the related subscription.

6.5 Referenced structured data types

6.5.1 Introduction

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

6.5.2 Type: TimeStamp

This type represents a time stamp.

Table 6.5.2-1: Attributes of the TimeStamp

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

6.6 Referenced simple data types and enumerations

6.6.1 Introduction

6.6.YY Type: <TypeNameYY>

7 API definition

7.1 Introduction

7.2 Global definitions and resource structure

7.w Resource: <ResourceNameW>

7.w.1 Description

7.w.2 Resource definition

7.w.3 Resource methods

7.w.3.1 GET

7.w.3.2 PUT

7.w.3.3 PATCH

7.w.3.4 POST

7.w.3.5 DELETE

Annex A (informative): Enabling MEC App providers to access MEC federation services

A.1 Introduction

According to ETSI GR MEC 035 [i.2], a "*MEC federation is a federated model of MEC systems enabling shared usage of MEC services and applications*". This definition is based on the need to standardize features taking into account the GSMA Operator Platform Group (OPG) OP Telco Edge requirements contained in [i.1].

In further detail, in [i.1], a number of interfaces are described, among which:

(i) the NBI connecting an application provider to an OP instance; and

(ii) the E/WBI connecting two OP instances.

Definitions of these two particular interfaces, as documented in [i.1], need to be considered by ETSI MEC when specifying the federation enablement services. Correspondences of NBI and E/WBI interfaces to MEC federation functional entities and reference points are shown in the exemplary Figure A-1, where it is assumed that each OP instance refers to a federated MEC system and there is no MEFB entity present.

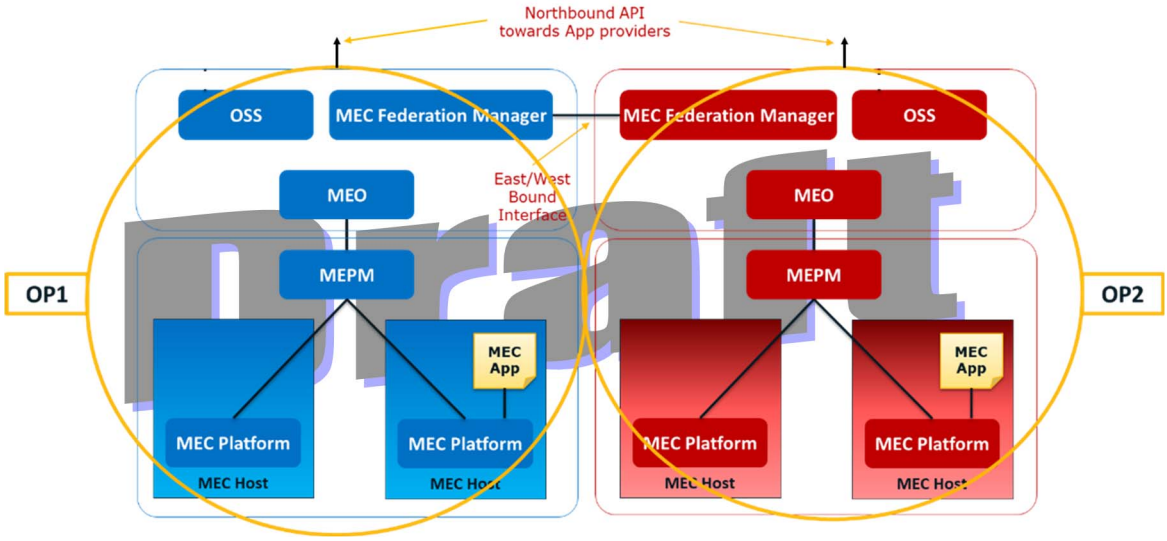


Figure A-1: MEC federation and exemplary correspondence to GSMA OP interfaces NBI and E/WBI [i.1]

NOTE: In the context of MEC federation, in the view of an implementation with 5G networks, both ETSI MEC and 3GPP standards should be considered, in a harmonized way.

In this scenario, enablement of MEC App providers to access the MEC federation in an interoperable manner is needed. This would imply the possibility for MEC App providers to offer MEC applications instantiated across the MEC federation, benefiting from all the resources (including MEC services) shared in the MEC federation, according to the authorization policies defined by the multi-party agreements among the business entities forming the MEC federation. In detail, what is needed is enabling an interoperable interface to MEC App providers accessing a MEC federation, with a list of available MEC services. Privacy and security of each MEC system the MEC federation is composed of needs to be ensured, considering that each MEC system which is part of the MEC federation has its own policy and need to expose information, customize access policies and capability to offer customized tools to MEC App providers. The E/WBI interface (along with the NBI interface) is an enabler for the access of MEC federation services by MEC App providers. In particular, the E/WBI enables MEC systems forming a MEC federation to be informed about the MEC App provider's requirement, information initially provided by the MEC App provider to a MEC system of a MEC federation through the NBI interface.

A.2 OP-NBI & OP-E/WBI API mapping to ETSI MEC specifications

A summary of the mapping of OP APIs to ETSI MEC specifications is provided in tables A.2-1 and A.2-2, with respect to the OP NBI and OP E/WBI, respectively.

Table A.2-1: Relationship between OP NBI APIs and ETSI MEC specifications

OP APIs	Description	SDO reference(s)	Stage 2 availability	Stage 3 availability	ETSI MEC corresponding APIs	Comments
Application onboarding and image management.						
Application Instance Management (Resource Life-Cycle Management)						
Telemetry						
Notifications						
Network Events						
Trouble Ticketing						
Application Resource Catalogue						
Ordering						
Charging						
Billing						
QoS Management						
Traffic Influence						
Managing Service availability in LADN						
Application relocation						
Confirm User Location						

Table A.2-2: Relationship between OP E/WBI APIs and ETSI MEC specifications

OP APIs	Description	SDO reference(s)	Stage 2 availability	Stage 3 availability	ETSI MEC corresponding APIs	Comments

Application onboarding						
Application Instance Management (Resource Life-Cycle Management)						
Telemetry						
Notifications						
Network Events						
Trouble Ticketing						
East/West Bound Interface Management						
Charging						
Billing						
QoS Management						
Traffic Influence						
Application Resource Catalogue						
East/West Bound Interface Management						
Availability Zone Information Synchronisation Service						
LBO Roaming (Monitoring)						
LBO Roaming (Authentication)						
Edge Node Sharing (resource onboarding & Management)						

Editor's note: tables A.2-1 and A.2-2 will be populated with content after the joint Workshop with GSMA OPG.

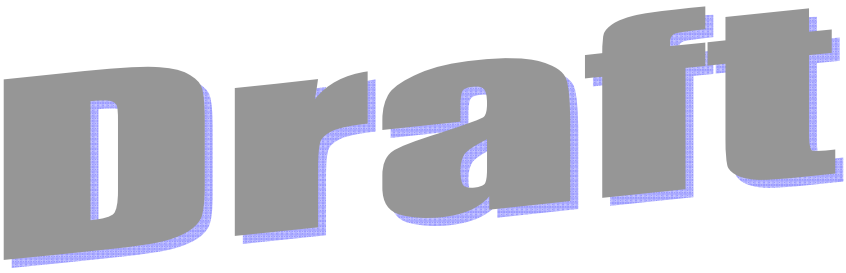
Editor's note: tables A.2-1 and A.2-2 could be also enhanced with relevant information on the 3GPP enablers referenced by ETSI standard, to exemplify the “packaging approach” proposed by ETSI MEC at the joint Workshop with GSMA OPG. This update can be, as well, e.g., part of the outcome of the joint Workshop, if all stakeholders find it convenient as a clarification for the reader.

Annex B (informative): Complementary material for API utilisation

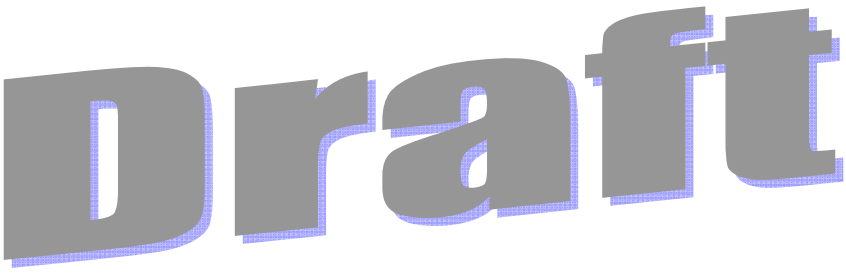
To complement the definitions for each method and resource defined in the interface clauses of the present document, ETSI MEC ISG is providing for the Application Mobility Service API a supplementary description file compliant to the OpenAPI Specification [i.7].

In case of discrepancies between the supplementary files and the related data structure definitions in the present document, the data structure definitions take precedence.

The supplementary files, relating to the present document, are located at https://forge.etsi.org/rep/mec/gs040-fed-enablement-api .

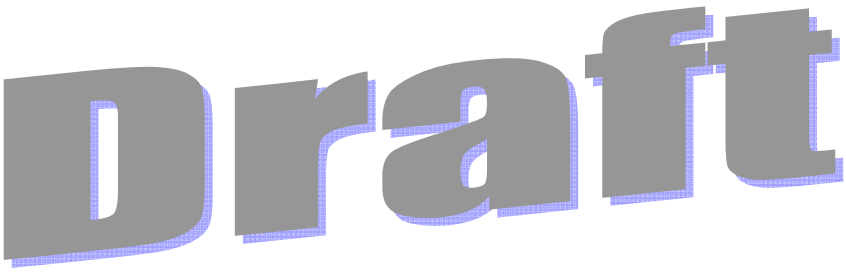


Annex (informative): Bibliography



Annex (informative): Change History

Date	Version	Information about changes
<Month year>	<#>	<Changes made are listed in this cell>



History

Document history		
V2.0.0	2021-06	Initial version of GS.
V2.0.1	2021-06	Implements document MEC(21)000260r2.
V2.0.2	2021-07	Implements document MEC(21)000268r5.
V2.0.3	2021-09	Implements documents MEC(21)000406r3, MEC(21)000407r2, and MEC(21)000408r2.
V3.0.4	2021-10	Implements documents MEC(21)000448r2, MEC(21)449r4. Corrects versioning.
V3.0.5	2021-12	Implements documents MEC(21)000373r5, MEC(21)000556, MEC(21)000547r2.
V3.0.5	January 2022	Clean-up done by ***editHelp!*** E-mail: mailto:edithelp@etsi.org
V3.0.6	January 2022	Early draft based on editHelp review, ready to go to MEC Open area.
V3.0.7	2022-01	Implements documents MEC(21)000548r9, MEC(22)000034r1, and MEC(22)000039r2.
V3.0.8	2022-02	Implements documents MEC(22)000049r1, MEC(22)000068r1, and MEC(22)000069.
V3.0.9	2022-03	Implements documents MEC(22)000100, MEC(22)000101r1, and MEC(22)000103r2.
V3.0.10	2022-07	Implements documents MEC(22)000288r3, MEC(22)000320r1, and MEC(22)000321.